

Fourth Annual Thermal and Fluids Analysis Workshop

(NASA-CP-10106) THE FOURTH ANNUAL
THERMAL AND FLUIDS ANALYSIS
WORKSHOP (NASA) 276 p

N93-13385
--THRU--
N93-13403
Unclass

G3/34 0129366

*Proceedings of a workshop
held at NASA Lewis Research Center
Cleveland, Ohio
August 17-21, 1992*



Fourth Annual Thermal and Fluids Analysis Workshop

*Proceedings of a workshop
held at NASA Lewis Research Center
Cleveland, Ohio
August 17-21, 1992*



National Aeronautics and
Space Administration

Office of Management

**Scientific and Technical
Information Program**

1992

CONTENTS

	Page
APPLICATIONS OF CFD AND VISUALIZATION TECHNIQUES	
James H. Saunders, Susan T. Brown, and Jeffrey J. Crisafulli, Battelle, and Leslie A. Southern, Ohio Supercomputer Center	1
THERMAL ISSUES AT THE SSC	
Raj P. Ranganathan and Bui V. Dao, Superconducting Super Collider, Accelerator Systems Division	9
NUMERICAL STUDY ON MIXING OF SPRAYED LIQUID IN AN LNG STORAGE TANK	
Hiroyuki Uchida, Tatsuya Arai, Makoto Sugihara, and Mariko Nakayama, Ishikawajima-Harima Heavy Industries Co., Ltd.	31
ANALYSIS OF A SPACE EMERGENCY AMMONIA DUMP USING THE FLOW-NET TWO-PHASE FLOW PROGRAM	
J. Navickas, McDonnell Douglas Space Systems Company, and W.C. Rivard, University of Maine	37
RUNNING SINDA '85/FLUNT INTERACTIVE ON THE VAX	
Boris Simmonds, Sverdrup Technology MSFC Group	47
TRASYS FORM FACTOR MATRIX NORMALIZATION	
Glenn T. Tsuyuki, Jet Propulsion Laboratory, California Institute of Technology	71
A SIMPLE NODE AND CONDUCTOR DATA GENERATOR FOR SINDA	
Ronald R. Gottula, GENCORP Aerojet	83
DEVELOPMENT STATUS OF SINDA/FLUNT AND SINAPS	
Brent A. Cullimore and Steven G. Ring, Martin Marietta Astronautics Group, and Eugene K. Ungar, NASA Johnson Space Center	95
A SINDA THERMAL MODEL USING CAD/CAE TECHNOLOGIES	
Jose A. Rodriguez, and Steve Spencer, Sverdrup Technology, Inc.	109
A COMPRESSIBLE BOUNDARY LAYER ALGORITHM FOR USE WITH SINDA '85	
Barbara Sakowski, and Douglas Darling, NASA Lewis Research Center, and Allan van de Wall, Case Western Reserve University	117
THE PROTEUS NAVIER-STOKES CODE	
Charles E. Towne, Trong T. Bui, Richard H. Cavicchi, Julianne M. Conley, Frank B. Molls, and John R. Schwab, NASA Lewis Research Center	131
ANALYSIS OF FLUID/MECHANICAL SYSTEMS USING EASY5	
Robert W. Clark, Jr. and Scott D. Arndt, McDonnell Douglas Space Systems Co., and Eric A. Hurlbert, NASA Johnson Space Center	155

ACOUSTIC MODES IN FLUID NETWORKS C.D. Michalopoulos, Robert W. Clark, Jr., and Harold H. Doiron, McDonnell Douglas Space Systems Co.	169
THE PROGRAM FANS-3D (FINITE ANALYTIC NUMERICAL SIMULATION 3-DIMENSIONAL) AND ITS APPLICATIONS Ramiro H. Bravo, Tri-State University, and Ching-Jen Chen, University of Iowa	187
ONE-DIMENSIONAL TRANSIENT FINITE DIFFERENCE MODEL OF AN OPERATIONAL SALINITY GRADIENT SOLAR POND Michael C. Hicks, NASA Lewis Research Center, and Peter Golding, University of Texas at El Paso	207
AEROSPACE APPLICATIONS OF SINDA/FLUINT AT THE JOHNSON SPACE CENTER Michael K. Ewert, NASA Johnson Space Center, Phillip E. Bellmore, McDonnell Douglas Space Systems Co., and Kambiz K. Andish and John R. Keller, Lockheed Engineering & Sciences Co.	221
TWO-PHASE/TWO-PHASE EXCHANGER SIMULATION ANALYSIS Rhyn H. Kim, The University of North Carolina at Charlotte	257
A SINDA '85 NODAL HEAT TRANSFER RATE CALCULATION USER SUBROUTINE Derrick J. Cheston, NASA Lewis Research Center	271

Applications of CFD and Visualization Techniques

James H. Saunders, Susan T. Brown, and Jeffrey J. Crisafulli
Battelle
Columbus, Ohio

Leslie A. Southern
Ohio Supercomputer Center
Columbus, Ohio

Introduction

Computational fluid dynamics (CFD) and data animation are powerful tools for understanding and solving complex engineering problems. The large data sets generated by time-dependent simulations can be dramatically illustrated with computer animation, often readily revealing the physics of the flow field.

In this paper, three applications are presented to illustrate current techniques for flow calculation and visualization. The first two applications use a commercial CFD code, FLUENT, performed on a Cray Y-MP. The results are animated with the aid of data visualization software, apE. The third application simulates a particulate deposition pattern using techniques inspired by developments in nonlinear dynamical systems. These computations were performed on personal computers.

Details of the simulations are presented elsewhere [refs. 1, 2, 3]. In this paper, we focus on visualization of the data.

Air Flow Within Air Conditioned Rooms

In the first application, we simulated the three-dimensional air flow in two air conditioned rooms connected by a doorway, with the goal of understanding the effects of blower fan on-time and return air vent placement on comfort level and air exchange within the room. Although real house flows are usually more complex, this simplified case represents the essential physics, and thus can be used to investigate basic flow patterns.

Figure 1 shows the rooms, which were 3.0 by 2.4 by 4.3 m and 3.9 by 2.4 by 4.3 m with a single 0.9 by 2.3 m door and insulated outside walls. The outdoor temperature was 90 F to simulate a hot summer day. The ceiling and floor were held at temperatures of 90 F and 73 F, respectively. 260 cfm of cool air at 55 F entered

the room through three inlet vents on the floor and exits through an outlet vent. The outlet vent could be located either on the floor or high on a wall. We considered two modes of fan operation: (1) running the fan only when the air conditioning was on, and (2) running the fan continuously. The air conditioner cycle of 15 minutes had an on-time of 6 minutes and an off-time of 9 minutes.

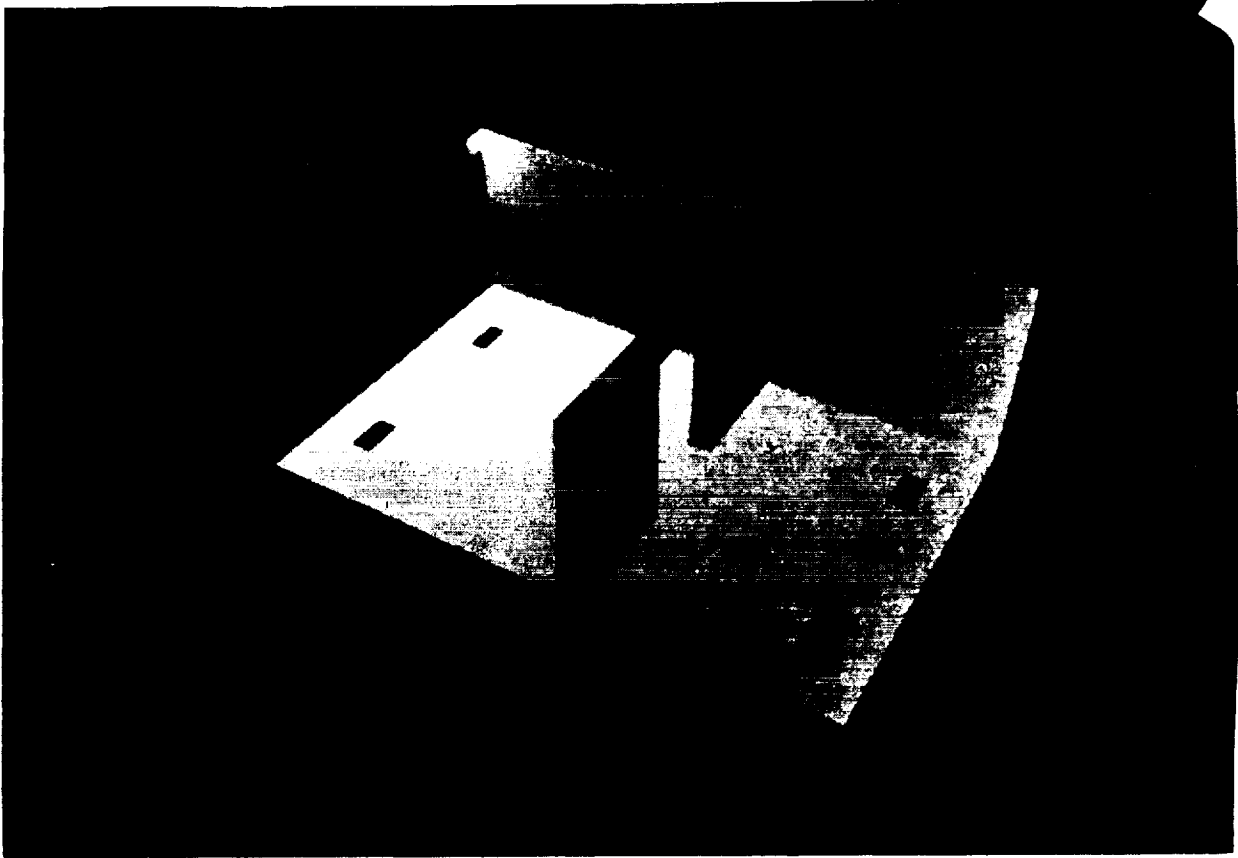


Figure 1. Animation of two air conditioned rooms

The room was modeled with 6061 nodes, using FLUENT, which solves the time dependent mass, momentum, and energy equations using a finite volume method. The temperature and velocity fields were then processed by apE to visualize the results. Three-dimensional objects and scenes were rendered by apE, using a scanline Z buffer approach to obtain photorealistic images that appropriately handled lighting, transparency and shading [ref. 4]. Polygonal iso-valued surfaces were constructed from the FLUENT data using a marching cubes algorithm [ref. 5]. For each timestep, three temperatures (77 F, 75 F, and 73 F) were illustrated with red, yellow, and blue isosurfaces, respectively.

The primary purpose of the visualization effort was to help characterize the air exchange within the two rooms as a function of fan on-time and outlet vent location. To visualize the air exchange process, a set of "glyphs" was used to mark the fluid. These massless particles, which track but do not interact with the flow, had two different shapes: pyramid shapes for existing room air and spherical shapes for the air entering through the vents as shown in Figure 2. For the glyphs to track the flow accurately, they had to interact with the velocity data generated by FLUENT. A special facility was written to perform this function in apE. The glyphs were color coded to indicate the local air temperature.

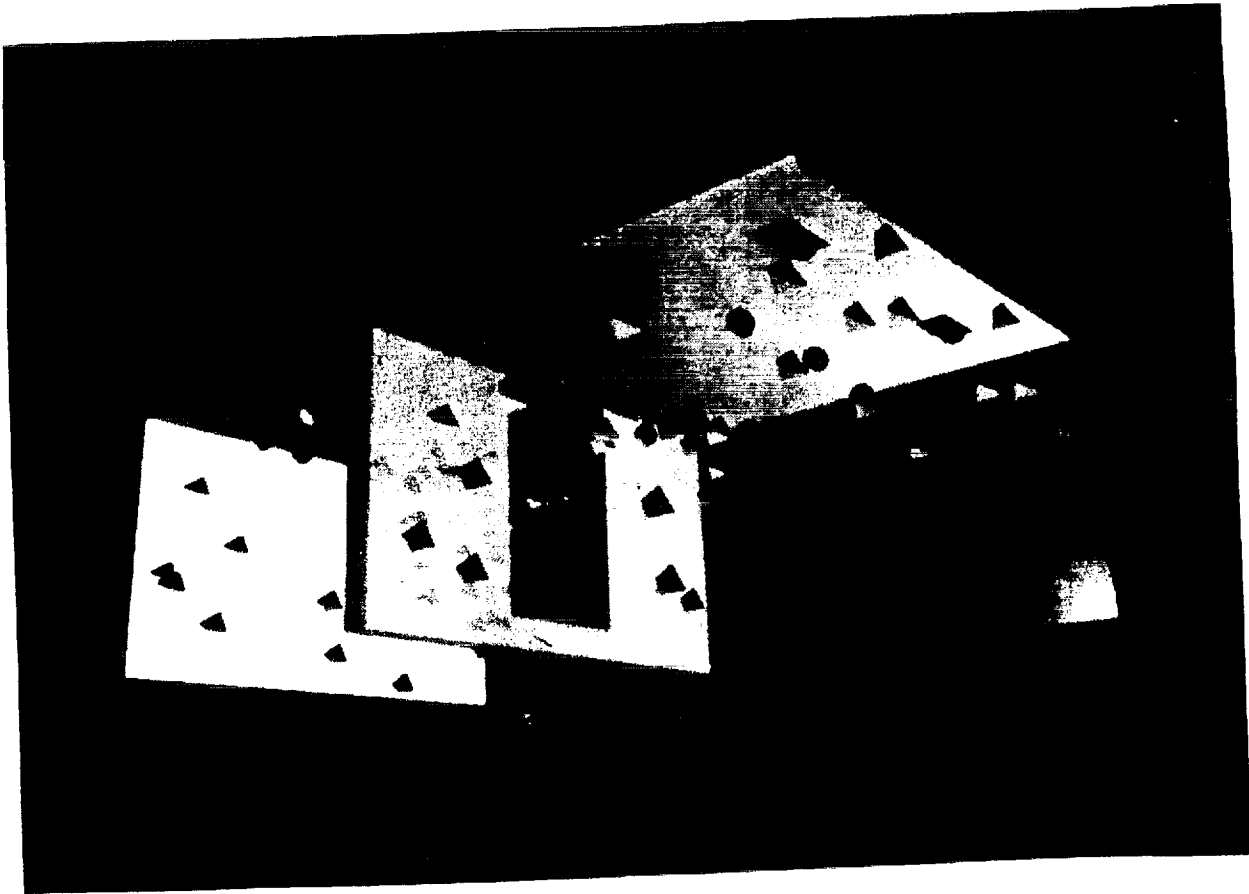


Figure 2. Visualization of the air exchange process: pyramid glyphs represent existing room air, spheres represent entering air

With the outlet vent on the floor and the fan running either intermittently or continuously, the isotherms are very flat, indicating poor mixing within the rooms. The glyphs clearly showed that the air short-circuited from the inlet vents to the outlet; the primary air flow, which was cool and dense, remained close to the floor and exited through the outlet vent without appreciable mixing with the older air in the room.

Placing the outlet high on the wall solved the poor air-exchange problem. The glyphs showed that the air flowed through a larger portion of the room volume as it passed to the outlet vent.

Indoor Flammable Plumes from CNG Leaks

Buses are often stored and maintained in large transit facilities, which may hold a large number of buses. A concern with natural gas-fueled buses is that a leak could create a flammable atmosphere in the transit building. Knowledge of the size of the plume for representative leaks is very important for developing future ventilation standards.

We analyzed the dispersion of leakage plumes inside a typical transit building that was 119 m by 108 m by 5.5 m high. During our simulation, the building was fully occupied with parked buses and all doors were shut. The ventilation system was on and operated at a rate of 5 air changes per hour.

Two leak scenarios were investigated:

1. A rapid leak corresponding to a ruptured fuel manifold line connecting the CNG cylinders or the failure of a pressure relief device.
2. Slow leaks from a poorly fitting fuel line connection. The leakage rate was up to 2.0 g/s.

Because of symmetry, one-fourth of the room was modeled with FLUENT using a grid of 12,000 cells. The effect of using a coarse grid on a flow with a wide range of geometrical length scales was assessed with some preliminary calculations. We found that leaking gas that entered the region between buses was strongly driven toward the ceiling by buoyancy forces. The details of the flow under or within buses were not important in determining the overall evolution of the plume.

The flammable concentration was tracked in time using apE. Two iso-valued surfaces were constructed; one represented the minimum flammable concentration, and the other represented the maximum. Transparency property effects were used for the maximum isosurfaces, so the flammable region was clearly depicted. The extent of the plume as a function of time was dramatically displayed with animation. Figure 3 shows a gray scale rendition of the fast-leakage-rate plume. The region between the dark and light surfaces of the plume represents the volume of the building with a flammable concentration of gas.

Particulate Deposition in Flow Systems

Particulate deposition and plugging in flow systems are important in a variety of industrial applications. We have simulated deposition in high-velocity gas

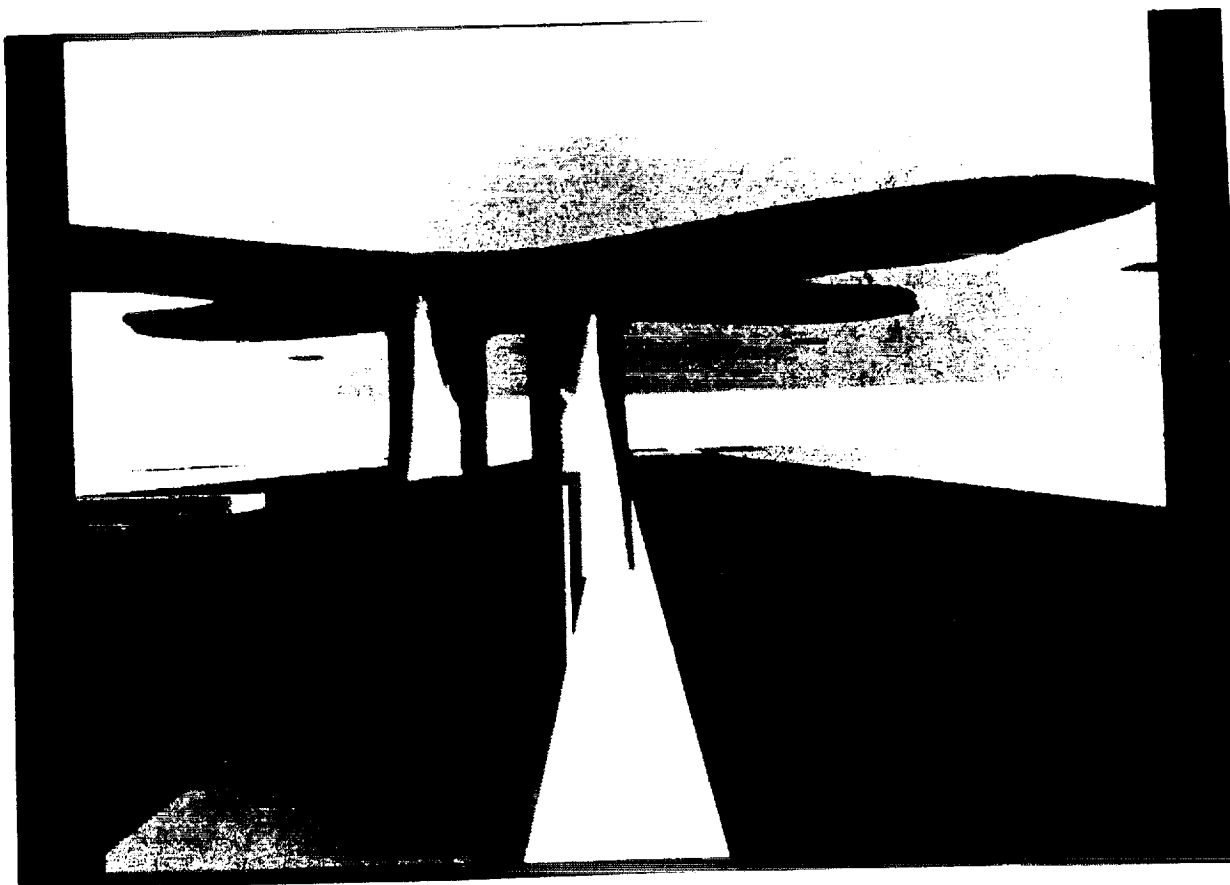


Figure 3. Rendition of fast-leakage rate indoor plume

flows where the flow is normal to a porous plate or collecting surfaces. Under these conditions, the particles travel in essentially straight lines without lateral diffusion or response to changes in the direction of the flow streamlines. Particles may collide with any surface they encounter. Because of the high velocity and high particle loadings, the deposit layer grows rapidly.

We modeled this process by tracking individual particles moving on a two-dimensional lattice as they form the deposit layer. Rules based upon the microphysics of the gas-particle-surface interactions determine whether a particle sticks to a deposit site, misses it, or bounces off. The computer algorithm displays the result on a high-resolution monitor, so that the development of the deposit can be observed continuously.

Our initial motivation for developing this technique came from studies on diffusion limited aggregation [ref. 6] and later from studies on ballistic deposition

[ref. 7]. Near the end of our work we became aware of other work on similar deposition models [ref. 8] with extensions shown in [ref. 9].

Figure 4 shows the results of simulating particles depositing on a porous plate. Particles, which are assigned one or more pixels on the graphics screen, are released at a random location above the deposit layer and are tracked as they move in a straight line toward the deposit. Deposited particles are shown as colored pixels on the screen. At each timestep, the algorithm examines pixels that are in and near the immediate path of the particle. If a collision with a deposited particle is imminent, the sticking probability is computed for that set of circumstances. If the collision will be a frontal collision, the particle may either stick or bounce. If the collision involves the sides or corners of the deposit, then the particle may stick or pass by.

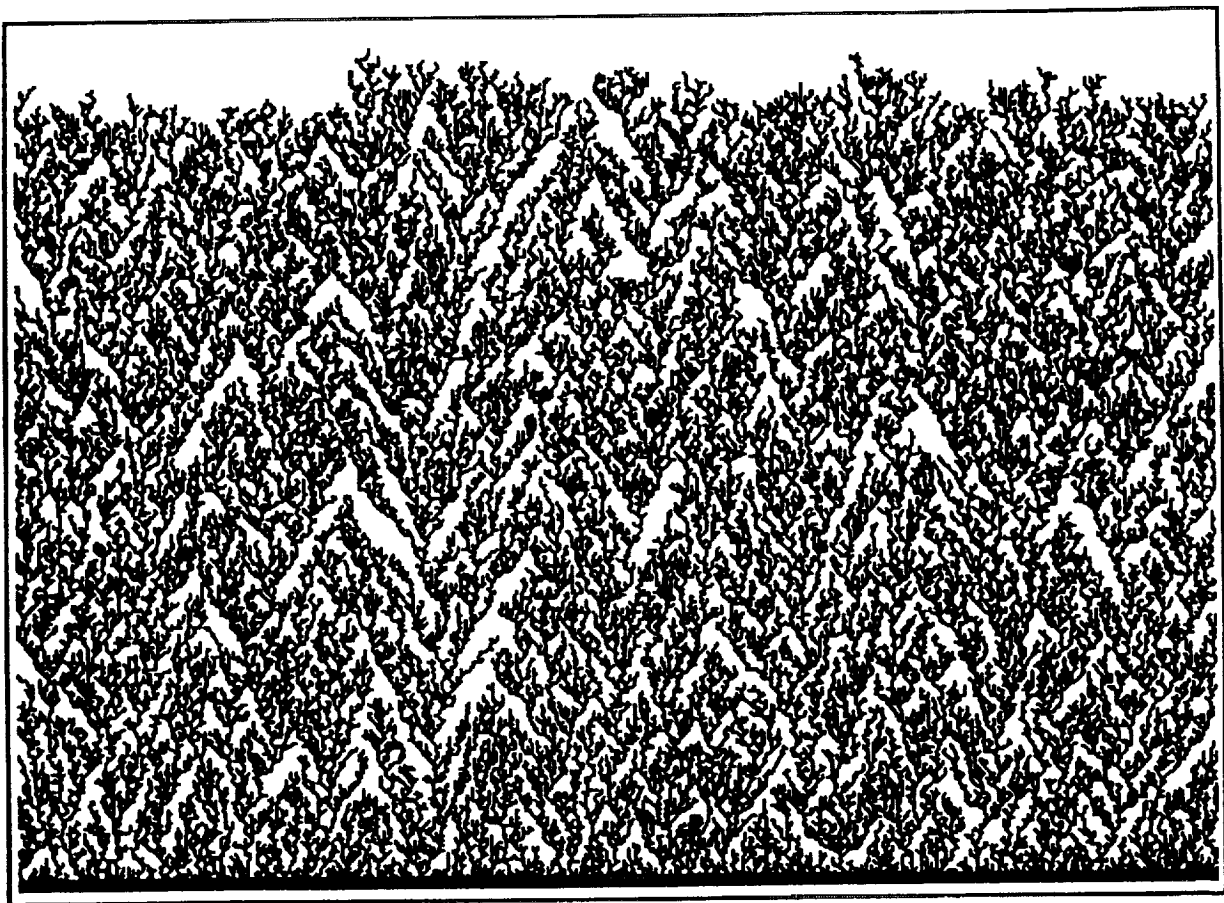


Figure 4. Simulation of particles depositing on a porous plate

Because of the continuously updated graphic display, the effects of rule changes can quickly be seen, allowing the researcher to evaluate the ramifications of

the assumptions and develop an understanding of the role of the microphysics on the formation of the resulting structure.

Our work has shown that the resulting deposit structure is sensitive to the form of the rules [ref. 5]. However, rules can be estimated from the detailed microphysics and future research should focus on extending this ability. For instance, in high speed flows, lateral dendritic growth may be strongly limited by shear-induced breakage of the dendrites. Rule selection should therefore be guided by careful comparisons of the predicted morphology of the deposit structure with detailed experimental measurements.

Conclusions

Visualization has been shown to be an important part of three engineering research problems using hardware ranging from supercomputers to personal computers. In the room ventilation example, the visualization revealed the impact of vent placement on air mixing in the two rooms. The visualization of the flammable gas plume in the transit building analysis gives the ventilation engineer a much clearer indication of potential weaknesses in the ventilation scheme than would be possible with traditional techniques. The graphical display of particle deposition gives the researcher a unique perspective on the growth of particle beds and allows detailed investigations of the particle microphysics in these processes. These visualization techniques have many applications that dramatically increase the usefulness of scientific data.

References

apE is a trademark of The Ohio State University

FLUENT is a trademark of FLUENT, Inc.

1. Murphy, M. J., S. T. Brown and D. B. Philips, "Extent of Indoor Flammable Plumes Resulting from CNG Bus Fuel System Leaks," to be presented at the SAE 1992 Truck and Bus Meeting and Exposition, November 16-19, 1992.
2. Brown, S. T., J. J. Crisafulli, D. B. Philips, L. A. Southern and D. F. Knight, "Analysis of Vent Placement and Fan Operation for Air Conditioning in a Partitioned Room Using Advanced Visualization Techniques," submitted for presentation at ASHRAE Winter Meeting, Denver CO, 1993.
3. Saunders, J. H., J. J. Crisafulli and G. H. Stickford, "Analysis of Particulate Depositions and Plugging in Flow Systems," *ASME Fluids Engineering Conference*, Los Angeles CA, June 21-24, 1992.
4. Snyder, J. M. and A. H. Barr, "Ray Tracing Complex Models Containing Surface Tessellations," *Proceedings of SIGGRAPH 1987, Computer Graphics*, Vol. 21, No. 4, July 1987, pp. 119-128.

5. Lorensen, W. E. and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proceedings of SIGGRAPH 1987, Computer Graphics*, Vol. 21, No. 4., July 1987, pp. 163-170.
6. Witten, T. A. and L. M. Sander, "Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon," *Phys. Rev. Letters*, Vol. 47, No. 19, 1981, pg. 1400-1403.
7. Meakin, P., P. Ramanlal, L. M. Sander and R. C. Ball, "Ballistic Deposition on Surfaces," *Physical Rev. A*, Vol. 34, No. 6, 1986, pp. 5091-5103.
8. Tassopoulos, M., J. A. O'Brien and D. E. Rosner, "Simulation of Microstructure/ Mechanisms," *AIChE J.*, Vol. 35, No. 6, 1989, pp. 967-980.
9. Tassopoulos, M. and D. E. Rosner, "Simulation of Vapor Diffusion in Anisotropic Particulate Deposits," *Chem. Eng. Sci.*, Vol. 47, No. 2., 1992, pp. 421-443.

THERMAL ISSUES AT THE SSC**Raj P Ranganathan and Bui V Dao**

**Accelerator Systems Division
Superconducting Super Collider
2550 Beckleymeade Avenue
Dallas, TX 75237**

(Presented at the Fourth Annual Thermal & Fluids Analysis Workshop, Cleveland, OH, August 17-21, 1992. Sponsored by the Thermal & Fluids Analysis Branch, NASA Lewis Research Center. Also published in the proceedings of the meeting.)

1.0 ABSTRACT

A variety of heat transfer problems arise in the design of the Superconducting Super Collider (SSC). One class of problems is to minimize heat leak from the ambient to the SSC rings, since the rings contain superconducting magnets maintained at a temperature of 4 K. Another arises from the need to dump the beam of protons (traveling around the SSC rings) on to absorbers during an abort of the collider. Yet another category of problems is the cooling of equipment to dissipate the heat generated during operation. An overview of these problems and sample heat transfer results are given in this paper.

2.0 INTRODUCTION

Figure 1 gives a birds-eye-view of the Dallas/Fort Worth area and the SSC main underground tunnel. The 87 km circumference tunnel will contain two main rings of magnets and other components [1].

Figure 2 shows a cross-section of the main tunnel. Two counter-rotating beams of protons will travel inside the two main rings at nearly the speed of light. Collision of these beams under controlled conditions is expected to yield new sub-atomic particles that will unravel mysteries of the origins of the universe.

The purpose of this paper is to give an overview of some of the heat transfer problems that arise in the design of the SSC and present sample heat transfer results.

3.0 CLASSIFICATION OF HEAT TRANSFER PROBLEMS

The heat transfer problems encountered at SSC can be classified (based upon their applications) into the following categories:

3.1 Heat Leak

Minimization of heat leak from the ambient to the components of the SSC main and High Energy Booster rings (that span a total route length of 185 km). The interior of these components is maintained at 4 K, since they contain superconducting magnets. Therefore, minimizing heat leak can lower refrigeration costs. References [2] through [6] are a small sample of the vast amount research reported on this subject.

3.2 Beam Absorption

Absorption of the beam of protons by depositing the beam on to absorbers may be necessary during commissioning of the SSC rings or during an abort of the rings [7]. Under accident conditions the beam could be deposited on to the superconducting magnets itself [8]. In addition, during beam deposition on to target materials (for physics experiments), a similar heat transfer problem arises.

3.3 Cooling of Equipment

Different equipment generate heat during their operation. The heat must be carried away by an optimum cooling system that does not jeopardise the various operational specifications of the equipment.

4.0 HEAT LEAK INTO THE SPOOL PIECE

4.1 Description

Figure 3 shows one of the components of the SSC rings, the spool piece [1]. At the spool piece, cryogen (helium and nitrogen) lines enter and leave the rings. The spool piece also contains safety valves for the cryogen tubing and in addition performs numerous other functions.

The spool piece is also characterized by a complex geometry with numerous heat flow paths between the ambient and the 4 K interior. Therefore, accurately predicting heat leak into the spool piece is a challenging task.

At steady state, the flow of heat into the spool piece from the ambient will be carried away by cryogen flowing at three temperature levels, namely 80 K, 20 K and 4 K. The cryogen at 80 K is liquid nitrogen, at 20 K it is gaseous helium and at 4 K it is liquid helium. The maximum temperature rise in the 4 K cryogen line is fractions of a degree kelvin, while in the 20 and 80 K cryogen lines, it is about 8 K.

4.2 Model Assumptions and Solution Details

In order to estimate the heat leak from the spool piece, the following simplifying assumptions were made:

- 1) heat flow through conduction paths is steady and one-dimensional,
- 2) residual gas conduction across the vacuum spaces is accounted for,
- 3) thermal radiation across the vacuum spaces is taken into account,
- 4) thermo-physical properties are a function of temperature,
- 5) natural convection of the cryogen in valves is neglected.

Numerous other secondary assumptions were made for the convenience of the analysis but are omitted here in the interests of brevity.

The conduction calculations were made using integral tables, while the effects of thermal radiation was accounted for by

means of simple calculations based on more detailed and accurate calculations [9]. The effect of residual gas conduction was based on empirical expressions [10].

4.3 Results

Table 1 summarizes the heat leak results for one of the many spool piece variants at SSC. Of greatest concern is the heat leak at the 4 K level since refrigeration costs are higher at lower temperatures. Table 1 shows that the total heat leak into the 4 K cryogen is nearly 9 W with the copper instrumentation leads accounting for more than 5 W.

A detailed thermal resistance analysis [6,9], possibly using available software in the market, may improve the accuracy of the heat leak estimates. Research has shown that natural convection flows of the stagnant cryogen in the valves may contribute to heat leak [11,12]. This assumption may have to be relaxed. Efforts are also underway to obtain measurements of the heat leak. More details of the calculations reported here can be obtained by contacting the principal author of this paper.

5.0 BEAM ABSORBER

5.1 Description

Figure 4 shows a schematic of an iron absorber for the SSC linear accelerator (LINAC). The beam of protons has an elliptic cross-section, with 2 mm and 20 mm axes, and is incident on the front surface of the absorber. Within the absorber, the energy of the protons is converted into internal energy of the absorber through a sequence of physics processes, the theory about which is available elsewhere [13].

The calculation of the energy deposition rates is accomplished by the most recent version of the MARS software [14]. As the beam penetrates the absorber, it assumes a cone shape and the energy deposition is correspondingly over a three-dimensional conical region within the absorber. The apex of the cone is at the point of incidence on the front surface of the absorber and the axis of the cone coincides with the axis of the beam. At the axis of the beam, the energy deposition rate is 4 orders of magnitude greater than at a radial distance of 1 cm. Thus high thermal stresses can be expected at the axis.

Thus, the problem here is to determine the duration of beam deposition that is permissible without exceeding the peak temperature and stress limit for a given absorber material.

5.2 Model Assumptions and Solution Details

The assumptions made to determine the peak temperature and stresses in the absorber were:

- 1) heat conduction is transient and three-dimensional,
- 2) heat generation rates are a function of t , x , y and z ,
- 3) thermo-physical properties are temperature dependent.

The problem was solved using ANSYS and computer times of the order of 1 day were required on the HP-730 workstation for each calculation.

5.3 Results

Figure 5 shows the peak temperature and stresses versus time for an iron absorber core for two different beam deposition scenarios. The scenarios were determined based on physics considerations [15]. Peak temperature and von-mises stress limits of 500 C and 200 MPa were specified, keeping in view the melting point of iron which is approximately 1500 C [16] and the yield point of iron under tension which is approximately 200 to 500 MPa [17].

The operational scenario (solid lines on Figure 5), involves a 1 GeV (giga electron volts) beam of protons impinging on the absorber in a series of pulses. Each pulse has a 7 micro-second duration and the corresponding current over that duration is 25 mA. The frequency of the pulses is 10 Hz. Further, the pulses arrive at the absorber in batches of 7, spanning 0.7 seconds. Each batch of 7 pulses is separated by a 6.3 second time interval from neighbouring batches. Thus, there are 7 pulses every 7 seconds.

For the operational scenario, the beam can be deposited on the absorber for 16 hrs without exceeding the above limits. Therefore, for the given scenarios, an iron absorber should be sufficient. Details of this work will be published shortly.

6.0 LIQUID COOLED RF-CAVITY TUNER

6.1 Description

Figures 6a and 6b show typical low energy booster radio frequency (rf) cavity. The function of the cavity is to accelerate the proton beam to higher energy levels [1].

Figure 7 shows a sectional view of an rf-cavity tuner. There are 4 ferrite disks of 25 mm thickness each, separated by 5 mm spaces. (Note, Figures 6a and 6b show 5 ferrites, while Figure 7 considers a configuration with 4 ferrites.) Coolant flows through the spaces between the ferrites to dissipate the heat generated in the ferrites, and in the walls of the tuner housing during the operation of the rf-cavity. Note the location of the coolant inlets and exits on Figures 6a through 7. The coolant inlets are diametrically opposite the exits.

The problem is to design an optimum cooling system that prevents high temperatures in the ferrites and the coolant. If the peak temperature in the ferrites approaches its curie temperature (125 C) then its magnetic properties are affected. Similarly, the peak temperature in the coolant should not approach its boiling point. In the case of one of the coolants considered here, the Galden Heat Transfer Liquid, the boiling point was 110 C. Details of this work are available elsewhere [18].

6.2 Model Assumptions and Solution Details

The following assumptions were made in the analysis [18]:

- 1) the heat transfer in the coolant and the flow field are steady, incompressible, three-dimensional and turbulent,
- 2) natural convection is included,
- 3) heat conduction in the ferrites is three-dimensional,
- 4) heat generation in the ferrites, coolant and copper is included,
- 5) the thermo-physical properties of the coolant and ferrite are constant,
- 6) geometric complexities neglected.

Due to reflective symmetry, two symmetry planes were identified normal to the axial and azimuthal directions, each of which bi-

sected the tuner to two mirror-image halves [18]. Thus the computational domain encompassed only one-fourth of the tuner shown on Figures 6a and 6b.

The problem was solved using the PHOENICS computational fluid dynamics (CFD) package. About 25,000 cells were used and computer times of several days was needed on the HP-730 workstation [18]. No comparisons of calculations with measurements have been made. When such data become available in the future, comparisons will be made. No grid dependence studies were made due to the enormous computer resources involved.

6.3 Results

Figure 8 illustrates a typical flow field on the axial-direction symmetry plane. Clearly the coolant prefers the path of least resistance along the annular passage between the ferrites and the tuner housing. Reducing the annular gap can induce the coolant to flow into the interior where cooling is needed.

The isotherms (also on the axial-direction plane of symmetry) of Figure 9 show a recirculation region. The peak ferrite temperatures were located there.

The isobars (near the side wall of the housing) of Figure 10 show that the bulk of the pressure drop in the fluid occurs near the exits.

In summary, the peak ferrite and coolant temperatures were sensitive to: coolant flow rate, coolant inlet temperature, inlet and exit areas, number of inlets and exits and the annular gap [18]. The calculations also indicated that natural convection effects played an important role in lowering the local temperatures in the coolant and ferrite[18,19].

The sensitivity studies helped influence the design of the cooling system.

7.0 SOLID COOLED RF-CAVITY TUNER

7.1 Description

A solid-cooled version of the rf-cavity described above was evaluated. Disks of Beryllium Oxide (BeO) or Aluminum Nitride (alride) were placed in the coolant spaces between the ferrites

(Figure 7) as shown on Figure 11 [20]. Five ferrite disks are used in this case (Figure 11) versus four in the earlier liquid cooled case. Good thermal contact between the BeO (or alnide) disks and the ferrite disks was facilitated by having a film of goop (a glue) between them. The glue also helped reduce thermal stresses in the ferrites and BeO (or alnide). Details of this work can be found in Reference [20].

Due to the high thermal conductivity of the BeO (286 W/mC) or the alnide (170 W/mC) compared to that of the ferrite (5.5 W/mC), the heat generated in the ferrites was transported to the housing walls by the BeO and alnide. At the housing walls the heat is removed by flowing a coolant such as water within tubes brazed on the outside of the housing. Thus, the BeO and alnide disks serve as paths of low thermal resistance that carry the heat away from the ferrites [20].

The problem involves heat conduction through complex geometries, with dissimilar solids bonded together. Therefore, the peak temperatures and stresses in the solids have to be determined.

It is important that the peak temperatures in the ferrites not approach its curie temperature (125 C) and the peak stresses in the ferrites, BeO and alnide yield a safety factor of at least 3 compared with the strengths of the respective materials. Table 2 shows the relevant strengths of the different materials.

7.2 Model Assumptions and Solution Details

The following assumptions were made:

- 1) transient, three-dimensional heat conduction,
- 2) heat generation rates vary with time and radius,
- 3) properties are constant.

The problem was solved using ANSYS and cpu times of several hrs were required on the HP-730 workstation for each case.

7.3 Results

Figures 12a and 12b show the computational domain from two different views. Notice the ferrites, BeO, copper housing walls and copper ribs outside the housing. The complexity of the geometry is evident.

Table 2 shows steady state temperature and stress results obtained for a typical case. Use of BeO gave the lowest peak temperature and stresses for all the cases considered. Use of alnides came second while the non-use of either BeO or alnide had the highest peak temperatures and stresses. In summary, solid cooling the tuner using BeO was found to be feasible from a thermal and stress perspective. Details are available elsewhere [20].

8.0 REFERENCES

- 1) "Site-Specific Conceptual Design of the Superconducting Super Collider," Material prepared by the staff of the SSC Laboratory, Editors: J. R. Sanford and D. M. Mathews, Superconducting Super Collider Laboratory Report SSCL-SR-1056, July 1990.
- 2) McAshan, M., Thirumaleshwar, M., Abramovich, S., Ganni, V. and Scheidemantle, A., "84 K Nitrogen system for the SSC," published in Supercollider 4, Proceedings of the IISSC, New Orleans, February, 1992.
- 3) Nicol, N. H., "SSC 50 mm collider dipole cryostat single tube support post conceptual design and analysis," SSC report number SSCL-N-765, July 9, 1991.
- 4) Carcagno, R. H., McAshan, M. S., and Schiesser, W. E., "A helium venting model for an SSC half cell," published in Supercollider 3, Proceedings of the IISSC, Atlanta, 1991.
- 5) Boroski, W. N., Nicol, T. H., and Schoo, C. J., "Design of the multilayer insulation system for the Superconducting Super Collider 50 mm dipole cryostat," published in Supercollider 3, Proceedings of the IISSC, Atlanta, 1991.
- 6) Franks, D. E. and Pletzer, R. K., "The effect of vacuum gas pressures and species on internal heat leak in the SSCL magnet design," published in Supercollider 4, Proceedings of the IISSC, New Orleans, February, 1992.
- 7) Hauviller, C., Schonbacher, H., van Steenberg, A., "Beam dump absorber for the 400 x 400 GeV p-p superconducting large storage rings (lsr)," CERN report no. CERN-ISR-GE/79-4, 1979.
- 8) VanGinnekin, A., "Distribution of heat due to beam loss in energy doubler saver type superconducting magnets," Fermi Lab report number TM-685, 1976.

- 9) Pletzer, R., "Results of revised analysis of 50 mm dipole magnets with and without 4 K and 20 K MLI blanket," SSC memo number 925830HM, March 23, 1992.
- 10) Barron, R. F., Cryogenic Systems, Oxford University Press, New York, 1985.
- 11) Timmerhaus, K. D., and Gu Y., "Theoretical analysis of thermal acoustic oscillation systems," presented at the XVIII th Int. Congress of Refrigeration, Montreal, Quebec, Canada, August 10-17, 1991.
- 12) Kuzmina, A. G., "Thermoacoustic oscillations in cryostats: new results," Cryogenics, vol. 32, no. 1, pp. 11-19, 1992.
- 13) Patterson, H. W. and Thomas, R. H., "Accelerator Health Physics," Academic Press, Inc., New York, 1973.
- 14) Mokhov, N. V., "The MARS10 Code System: Inclusive Simulation of Hadronic and Electromagnetic Cascades and Muon Transport," Fermi National Accelerator Laboratory Report FN-509, March, 1989.
- 15) McGill, J., Bull, J., Shailey, R., and Wang, F., physicists at Superconducting Super Collider Laboratories, 1992.
- 16) Lide, D. R., "Handbook of Chemistry and Physics," 71 st Edition, CRC Press, Boston, 1990-1991.
- 17) "Handbook of Mathematical, Scientific, and Engineering Formulas, Tables, Functions, Graphs, Transforms," Staff of Research and Education Association, Dr. M. Fogiel, Director, Research and Education Association, New Jersey, 1984.
- 18) Ranganathan, R. P., "Tuner thermal analysis," presented at the rf Workshop, TRIUMPF, Vancouver, B.C., Canada, January 21-24, 1992.
- 19) Campbell, B., "SSC LEB Cavity Mechanical Design Considerations," presented at the rf Workshop, TRIUMPF, Vancouver, B.C., Canada, October 25-26, 1990.
- 20) Ranganathan, R. P., "Solid Cooled Tuner," presented at the rf Workshop, SSC Labs, Dallas, Texas, June 25, 1992.

Table 1: ASST SPR spool piece heat load summary.

	HEAT LEAK PATHS	LOCATION IN SPOOL PIECE (mW)		
		4 K	20 K	80 K
1	Floating Support	15	320	2103
2	Instrumentation Tuving (Lead End)	89	0	0
3	Cool-Down Valve Tubing	0	0	2849
4	4 K Relief Tubing & Cryogen Conv.	0	0	891
5	Correction Element Power Lead ASSY	1930	0	0
6	20 K Relief Tubing & Cryogen Conv.	0	109	804
7	Quench Line	28	95	0
8	Recooler Valve	165	0	556
	Stainless Steel Housing	[88]	[0]	[551]
	G-10 Housing	[6]	[0]	[45]
	UHMW Plug-In Stem	[71]	[0]	[0]
9	Recooler Valve Pressure Tubing	88	537	5042
10	Fixed Support	15	320	2103
11	Deflection Stops	24	0	0
12	Instrumentation Tubing (Middle)	174	0	0
13	Vacuum Barrier (With Cu Straps)	78	1573	10775
14	Instrumentation Tubing (Return End)	106	0	0
15	Beam Tube Pump Port	55	0	581
16	Radiation Between Shields	1	192	8441
17	Residual Gas Conduction	33	60	6
18	Interconnect Regions	0	42	38
19	99.99 % Cu Instrumentation Wires	5222	0	0
	J1 (8 22-AWG; 7 28-AWG)	[757]	[0]	[0]
	J2 (12 14-AWG)	[4067]	[0]	[0]
	JSPR (60 30-AWG)	[56]	[0]	[0]
	JRE1 (60 30-AWG)	[342]	[0]	[0]
20	Manganin Instrumentation Wires	7	1	2
	J3 (40 32-AWG)	[4]	[0]	[0]
	JVAC (36 32-AWG)	[3]	[1]	[2]
21	BPM Cables	120	0	0
22	Safety Leads	2	0	0
23	Opening Near Vacuum Barrier	201	0	0
TOTAL CALCULATED ASST HEAD LOAD		8353	3249	34191
24	Dynamic Heat Load	462	0	0
TOTAL CALCULATED ASST HEAT LOAD		8715	3249	34191
TOTAL BUDGETED COLLIDER HEAT LOAD		2288	11321	57710

Greg Cruse, John Nguyen, Raj Ranganathan, Andy Scheidemantle, Ken Schiffman
02/28/92

**Table 2: Thermal/stress results of solid cooled low energy
booster rf-cavity tuner**

1	No. of Alnide Disks	0	2	0
2	No. of BeO Disks	0	0	2
BEO				
1	Max T (° C)	-	-	59
2	Max Tension (MPa)	-	-	8
3	Tensile Strength (MPa)	-	-	151
4	Safety Factor	-	-	18.9
ALNIDE				
1	Max T (° C)	-	63	-
2	Max Tension (MPa)	-	32	-
3	Flexural Strength (MPa)	-	46	-
4	Safety Factor	-	1.4	-
FERRITE				
1	Max T (° C)	85	66	62
2	Curie T (° C)	125	125	125
3	Max Tension (MPa)	26	15	14
4	Tensile Strength (MPa)	39	39	39
5	Safety Factor	1.5	2.6	2.8
GOOP				
1	Max T (° C)	83	66	61
2	Max Operating T (° C)	150	150	150
3	Max Shear (MPa)	<< 1	<< 1	<< 1
4	Lap Shear Strength (MPa)	7	7	7
5	Safety Factor	>> 7	>> 7	>> 7
COPPER				
1	Max Stress (MPa)	25	25	24
2	Yield Strength (MPa)	69	69	69
3	Safety Factor	2.8	2.8	2.9

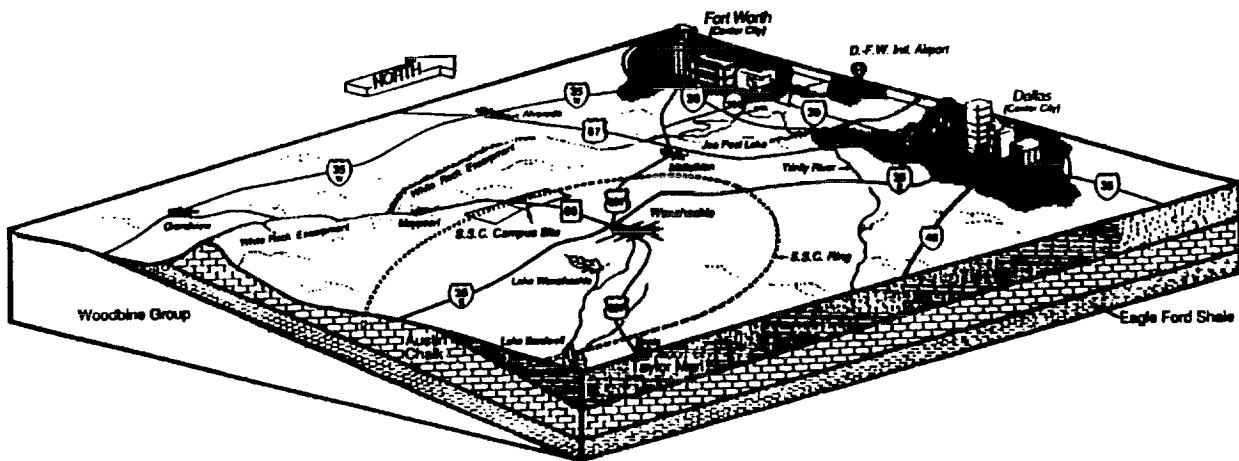


Figure 1: Topographic and geologic profile along ring circumference [1].

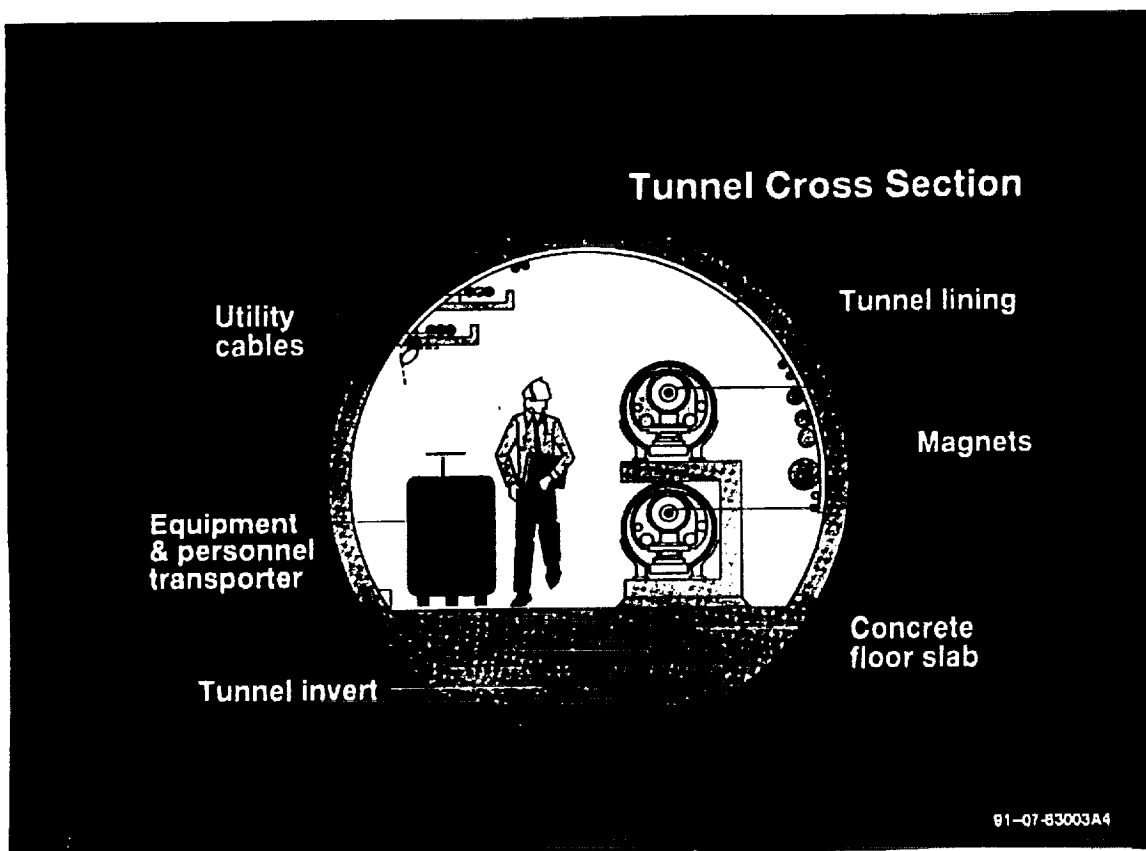


Figure 2: Cross section of the main SSC tunnel

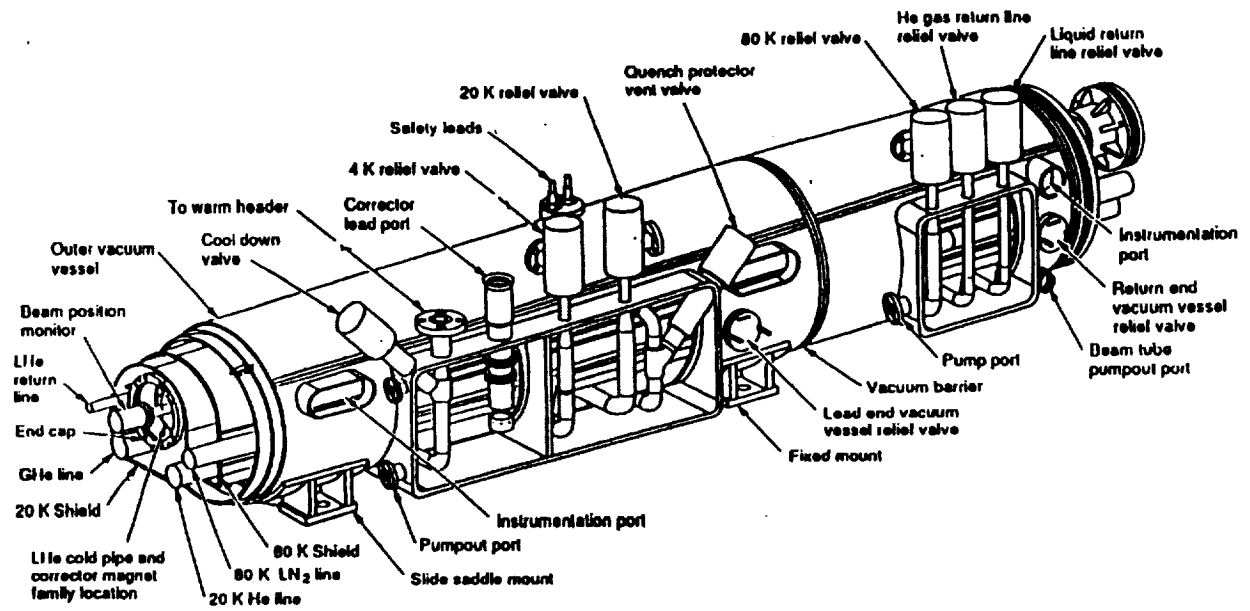


Figure 3: Standard spool piece

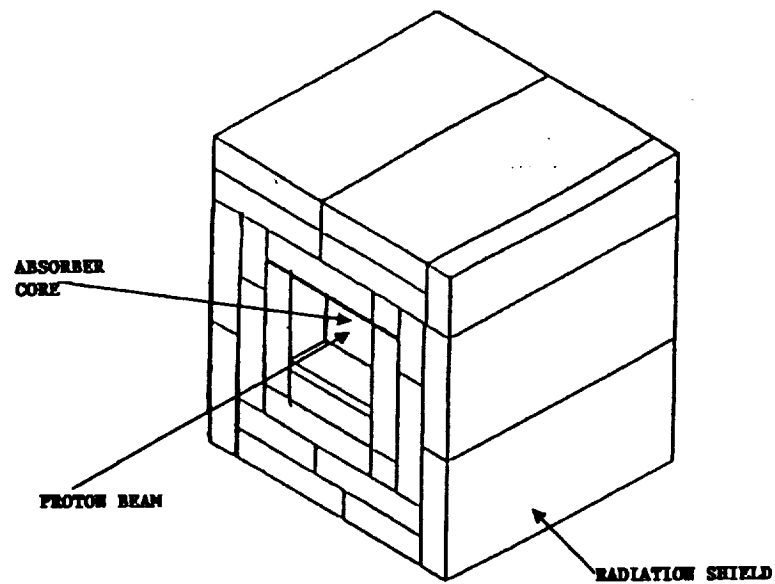
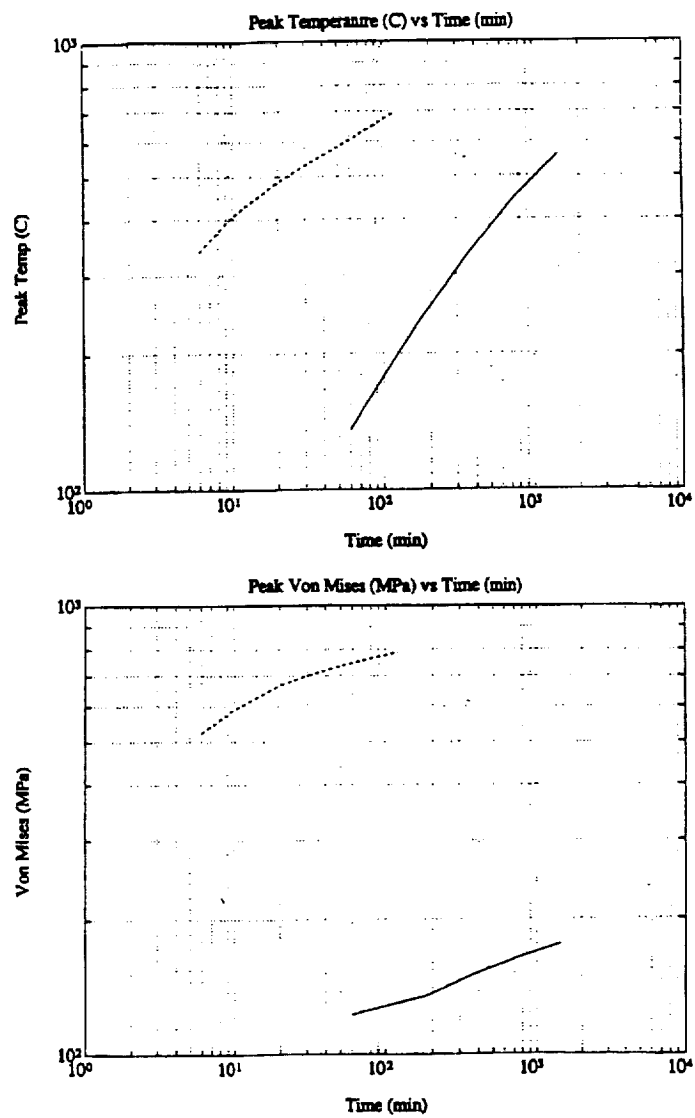


Figure 4: Schematic of a beam absorber



— 1 GeV, 25 mA, 7 mic-s pulse @ 10 Hz, 7 pulses / 7 s

- - 1 GeV, 50 mA, 35 mic-s pulse @ 10 Hz, continuous

Figure 5: Peak temperatures and stresses in the LINAC beam absorber

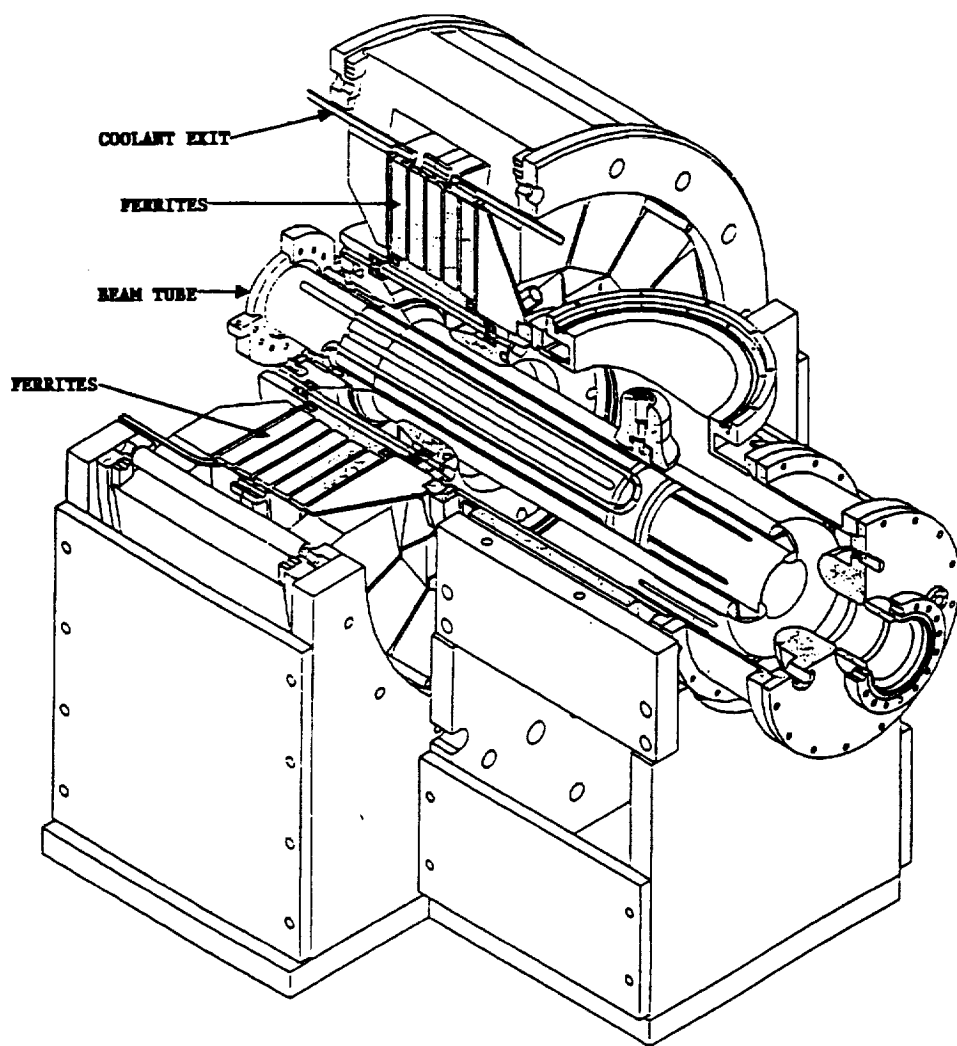


Figure 6a: Representative low energy booster rf-cavity

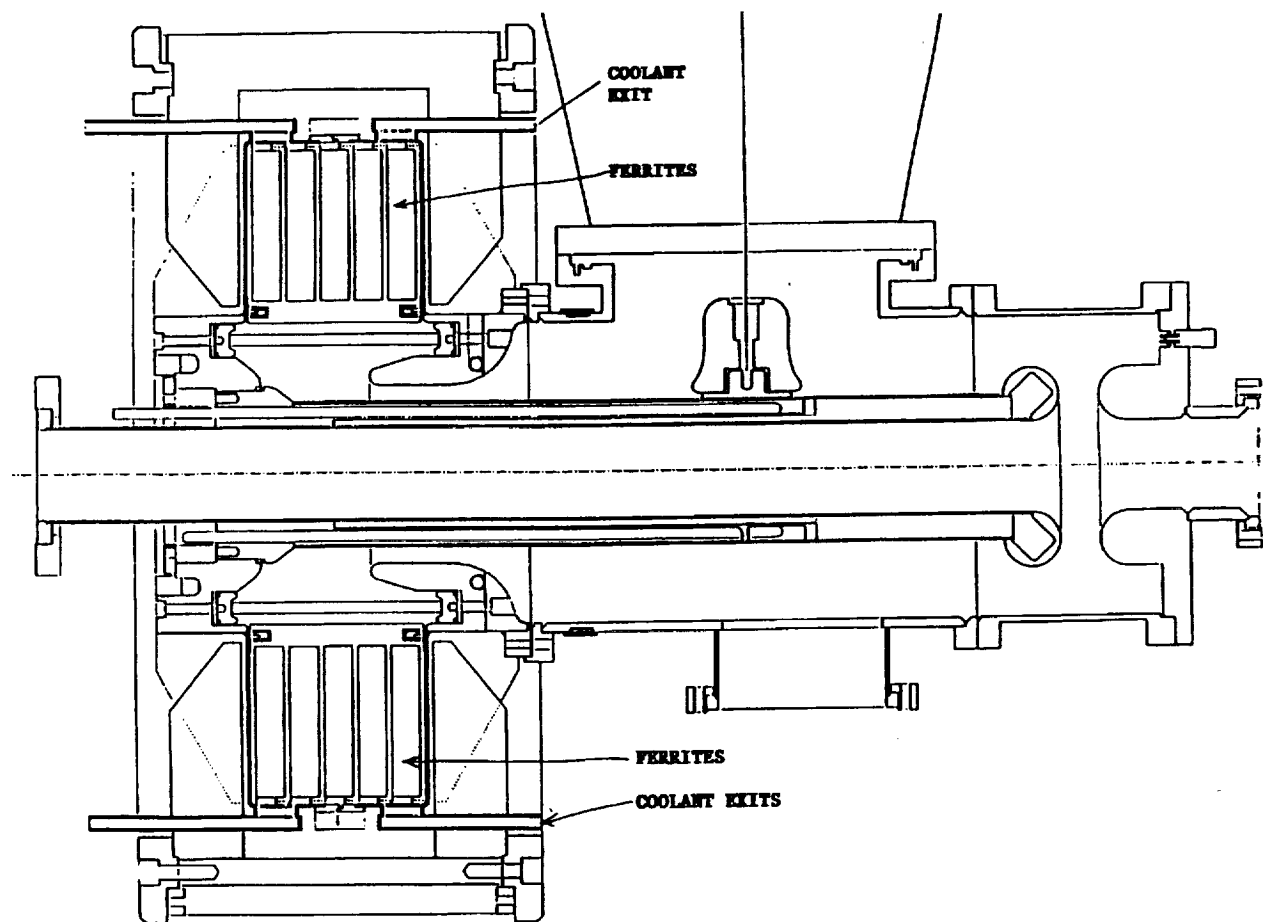


Figure 6b: Representative low energy booster rf-cavity -- sectional view

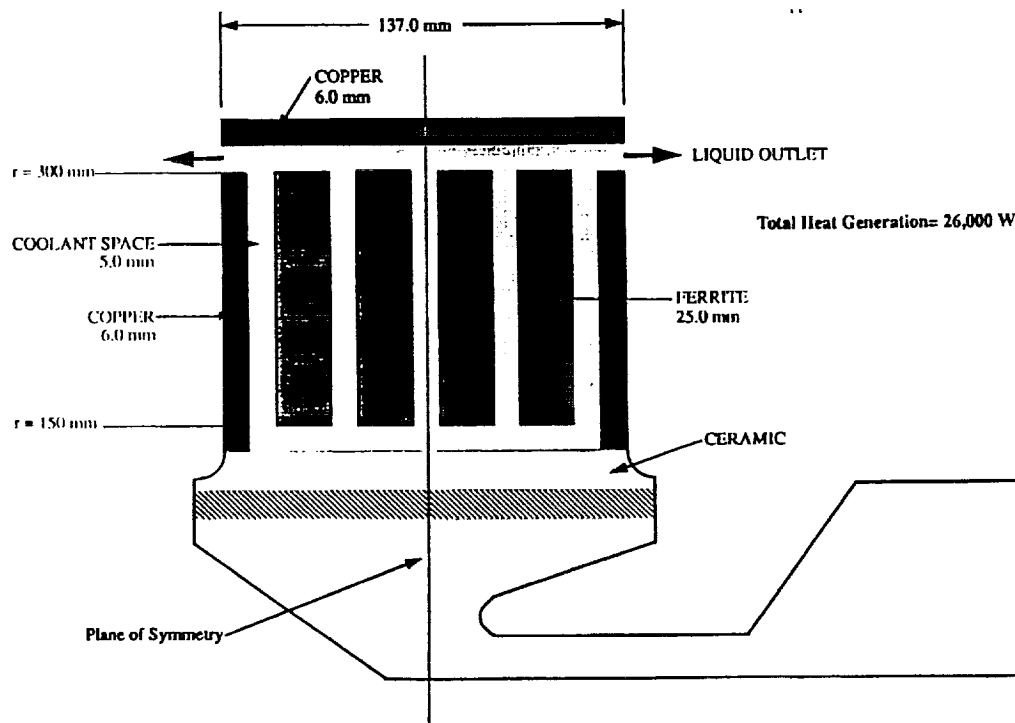


Figure 7: Sectional view of liquid-cooled low energy booster rf cavity tuner

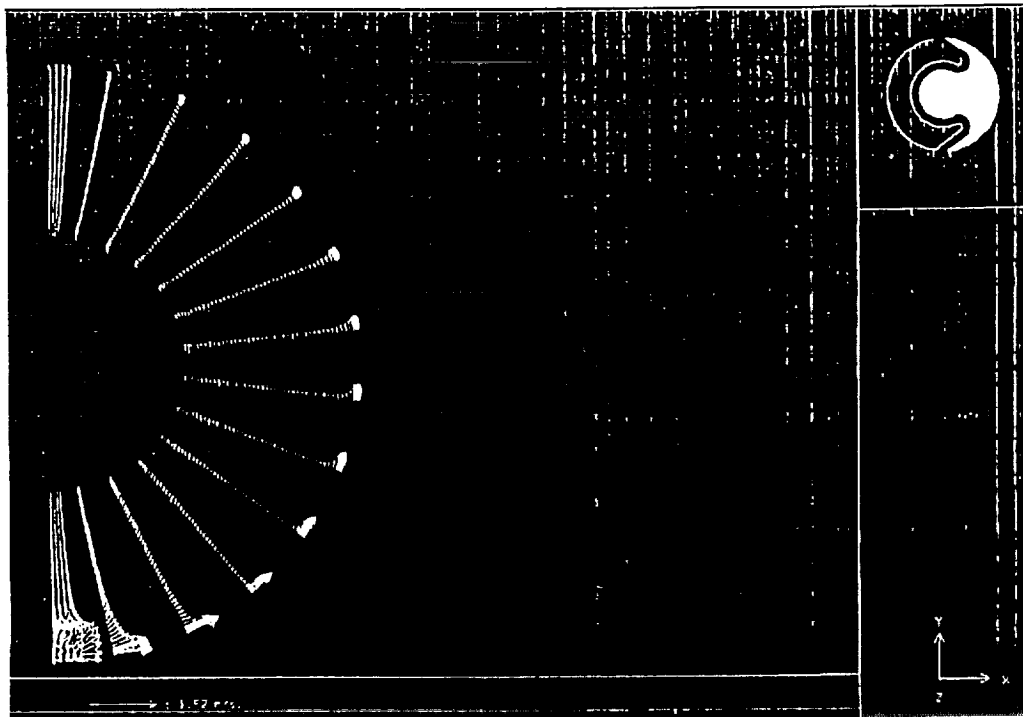


Figure 8: Velocity field on the axial direction symmetry plane

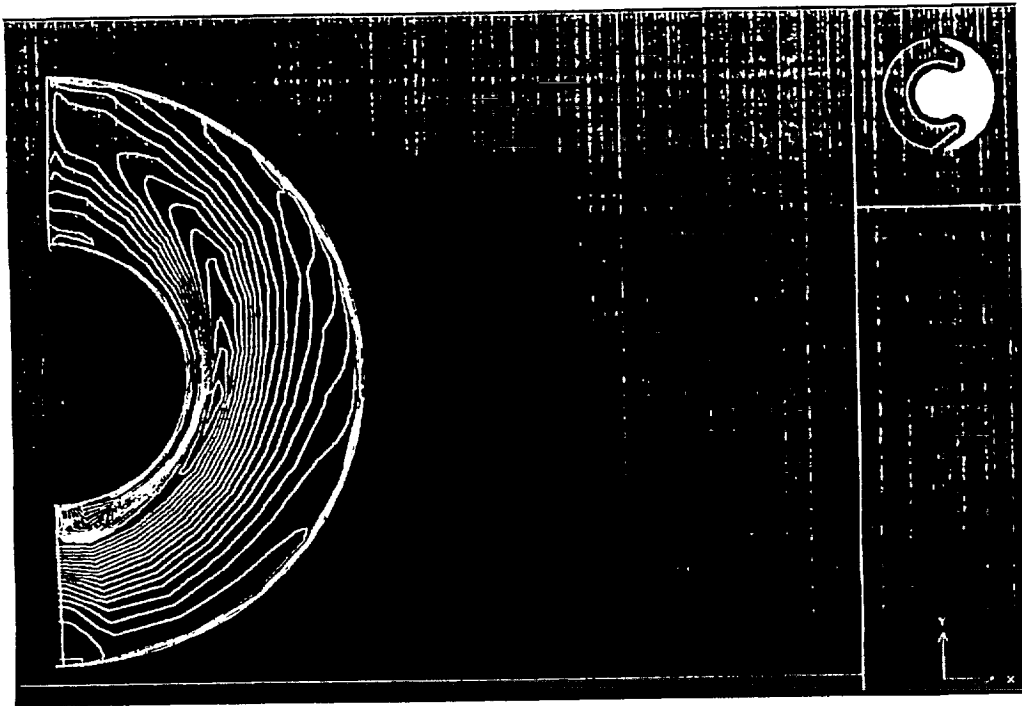


Figure 9: Coolant isotherms on the axial direction symmetry plane

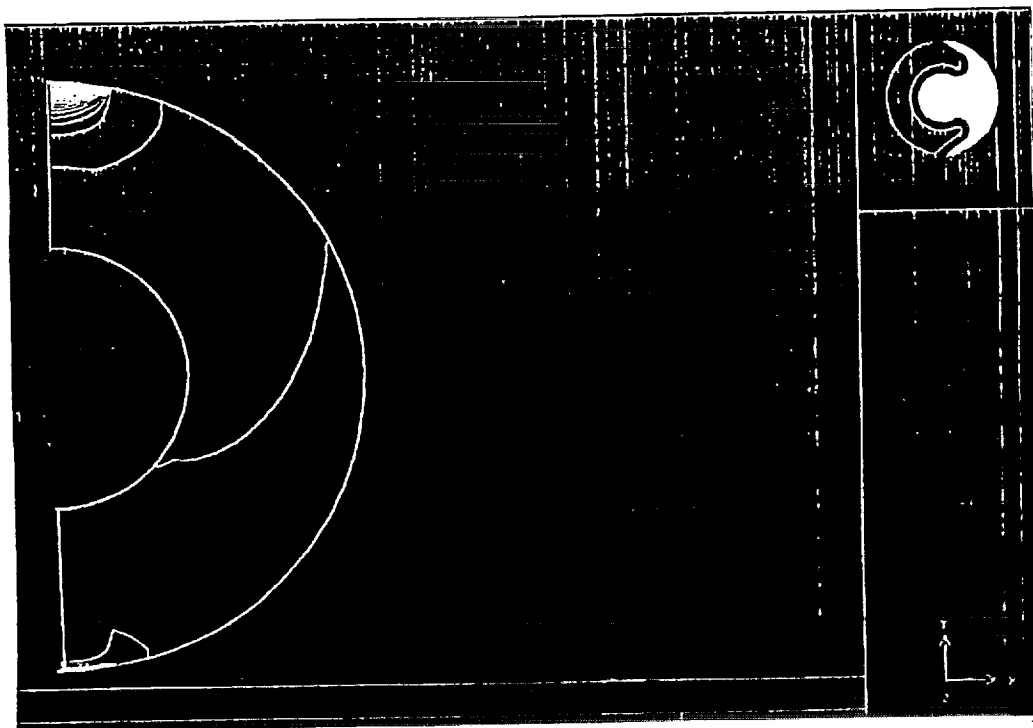


Figure 10: Isobars near the side wall of the housing

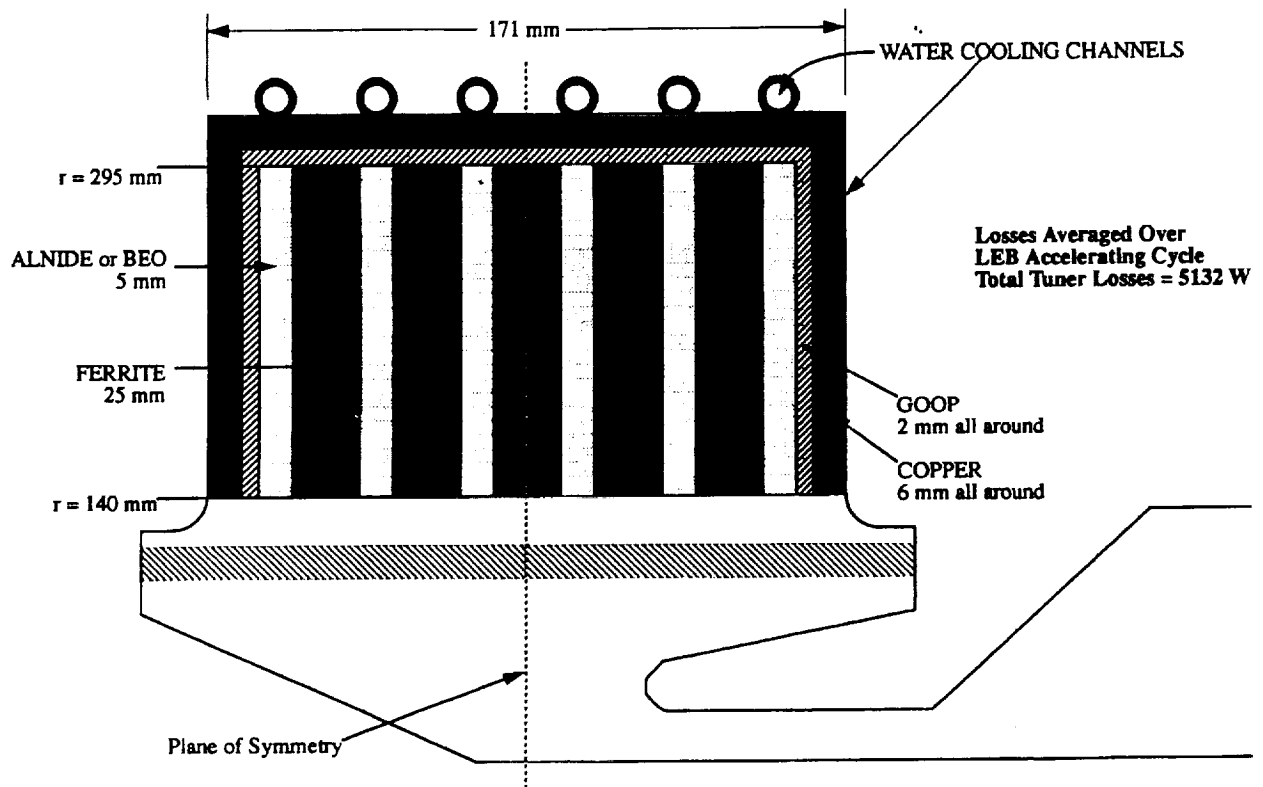


Figure 11: Sectional view of solid cooled low energy booster rf-cavity tuner

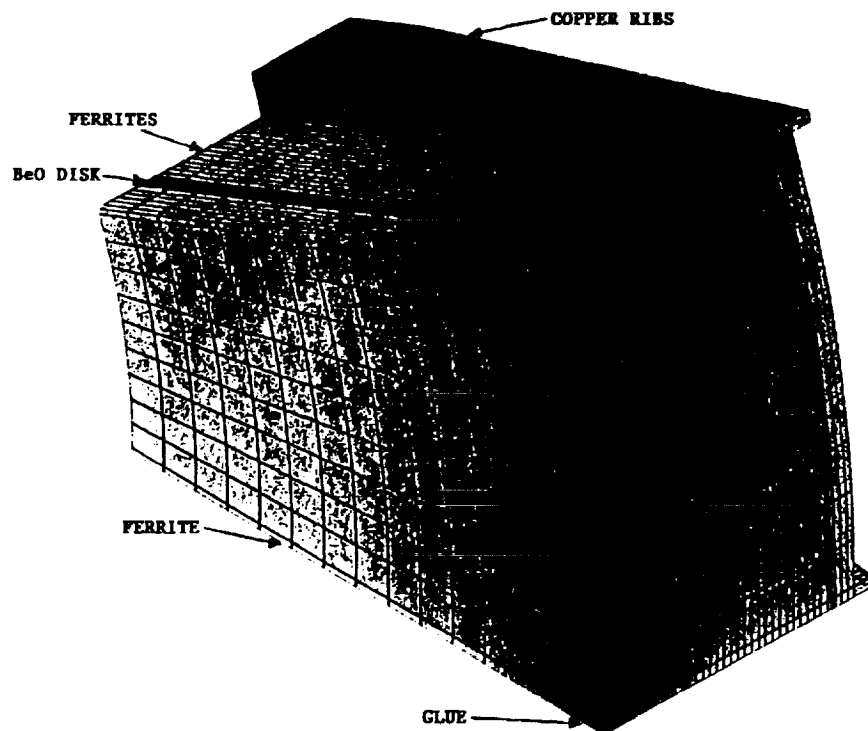


Figure 12a: Computational domain for the solid cooled low energy booster rf-cavity tuner - view 1

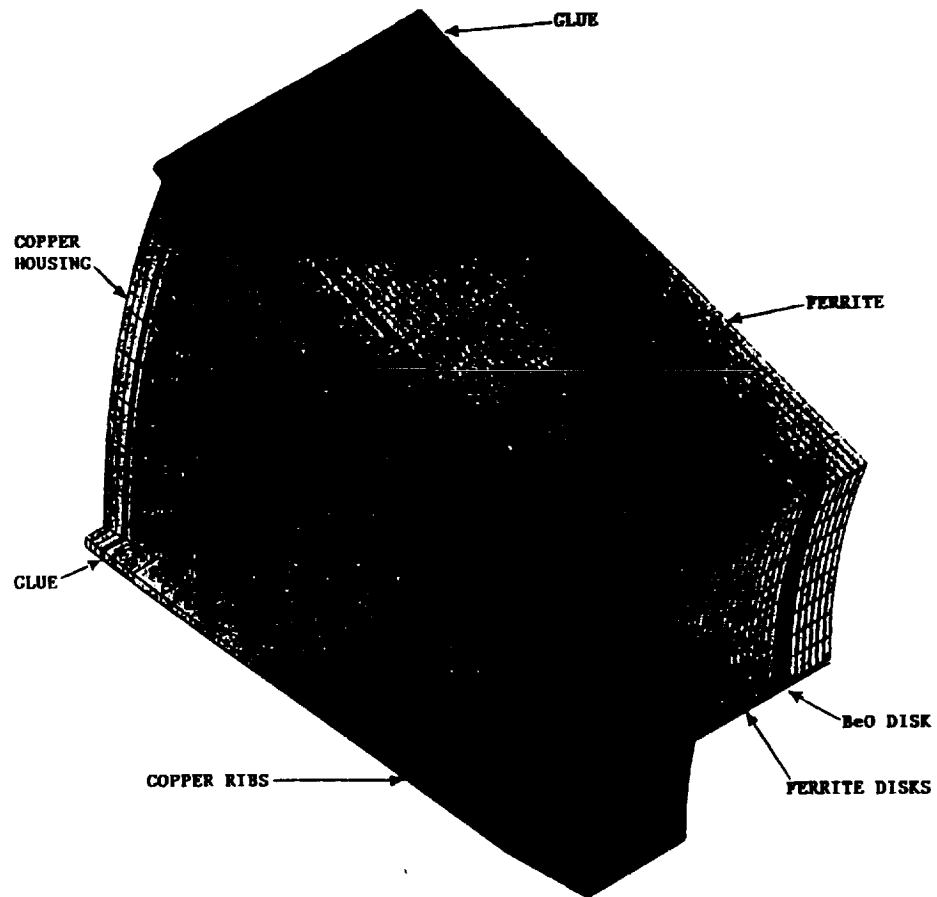


Figure 12b: Computational domain for the solid cooled low energy booster rf-cavity tuner - view 2

NUMERICAL STUDY ON MIXING OF SPRAYED LIQUID IN AN LNG STORAGE TANK

Hiroyuki Uchida, Tatsuya Arai
Makoto Sugihara and Mariko Nakayama
Ishikawajima-Harima Heavy Industries Co., Ltd.
Yokohama, Japan

SUMMARY

This paper presents a numerical method to simulate the mixing of heavier LNG sprayed on lighter layer. Numerical results for evolutions of flow field and density field are obtained in a rectangular computational domain which includes the vicinity of the liquid surface. At the surface boundary, uniform distributions of the fluid velocity and the density are assumed. Detail structure of flow caused by impingements of liquid drops are neglected. But, to trigger a realistic motion, a series of random numbers is employed. It is used as an initial distribution of the density near the surface. This method successfully gives a realistic simulation of the mixing process. Numerical result for mixing velocity shows good agreement with experimental data.

INTRODUCTION

Density of LNG varies according to its composition. When we receive LNG with different composition into a partially filled tank, they sometimes separate into two layers. This stratification should be avoided because it may cause the roll-over accident. A widely accepted way to receive heavier liquid into lighter layer is called bottom feed method, where these liquids are mixed by jet flow from a nozzle placed at bottom of the tank. In these years, top feed method is adopted at several power stations, where heavier liquid is sprayed on the lighter layer through a ring-header placed in the top space of the tank. This method is believed to be more reliable to receive heavier LNG.

Many researches have been done related to the mixing in the bottom feed method both theoretically and experimentally (refs. 1 and 2), but no reports are found for the top feed method. Some researches have been done on impingement of single drop against liquid surface (refs. 3 and 4), but there seems no research dealing with many drops and/or mass transport. In the present study, mixing process of the top feed method is investigated numerically. A color animation video will be presented at the meeting to show an evolution of the density distribution.

MODEL AND METHOD OF COMPUTATION

In the present analysis, temporal change of the distributions of velocity and density are solved in a rectangular computational domain. This domain includes the vicinity of the liquid surface. Therefore, only beginnings of the mixing can be analyzed in the present study. Governing equations employed here

are vorticity transport equation, stream function equation (Navier-Stokes equations) and mass transport equation. Buoyant force due to the density difference is modeled by using Boussinesq approximation. This system is governed by three nondimensional parameters: Grashof number Gr, Reynolds number Re and Schmidt number Sc. That is,

$$\begin{aligned} Gr &= g(\Delta\rho/\rho_0)L^3/\nu^2 \\ Re &= UL/\nu, \quad Sc = \nu/D \end{aligned}$$

where g denotes the acceleration of gravity, $\Delta\rho$ the density difference between heavier and lighter liquid, ρ_0 the density of the sprayed (heavier) liquid, L the reference length, U the receiving velocity, ν the kinematic viscosity and D the diffusion coefficient.

Uniform distributions of the inflow velocity and the density are assumed at liquid surface, i.e. top boundary. That is, flow due to the impingement of the drops are ignored because of small scale of the flow. A thin layer with uniform density is assumed to form near the surface as a result of quick mixing. However, a nonuniform initial distribution of density is given on a grid line just below the top boundary. A series of random numbers is used to make this nonuniformity. This trick enables to obtain realistically complex solution. Uniform distribution of the velocity is given at bottom boundary. Both side boundaries are modeled as no-slip wall.

The governing equations are discretised by using a finite difference method. The transport equations for the vorticity and the density are solved by an explicit time integration method. In the present problem, mass transport will be dominated by convection because of very small diffusion coefficient. To keep high accuracy for such complex flow, Kawamura-Kuwahara scheme (ref. 5) is used to approximate convection terms. The stream function equation is solved by using the ADI method in each time step.

Some essential input data used in the present computation are as follows: $Gr=1.1 \times 10^{10}$, $Re=47.4$, $Sc=794$, 40×120 grids, 20000 time steps. Peclet number $Pe (=ReSc)$ is 3.76×10^4 . The reference length $L (=200\text{mm})$ is a width of an experimental apparatus by which a visualization experiment was carried out. Physical properties are for brine. Computation time was about 3.5 hours on a computer FUJITSU FACOM VP-2100.

RESULTS AND DISCUSSION

In the early stage of computations, we did not use any artificial initial distribution of density. But there happened no convective mixing. It is a trivial solution with only diffusion. For the next trial, we gave a seed for an initial distribution of the density. Some small value was given at only grid point on the center line just below the surface. Figure 1 shows an evolution of the density distribution for initial 4 seconds. Because this is just a trial, the lower part of the computational domain was cut off. To avoid the complication of the figure, the contour line is plotted for 0.1 (nondimensional density) only.

In the beginning of the mixing, wavy motions appear in the vicinity of the liquid surface. These motions are similar to those seen in Rayleigh-Taylor instability (ref. 6). The characteristics of this wavy motion, such as wave

length, are determined spontaneously. The amplitudes of the waves increases gradually. One of the plume stands out from the others, and a mushroom-shaped plume forms. This plume reaches bottom and spreads. The density distribution is perfectly symmetrical and not very complex in contrast with the following result. We have not seen such simple and beautiful patterns in the experiment. This simplicity may come from the unrealistic boundary or initial condition.

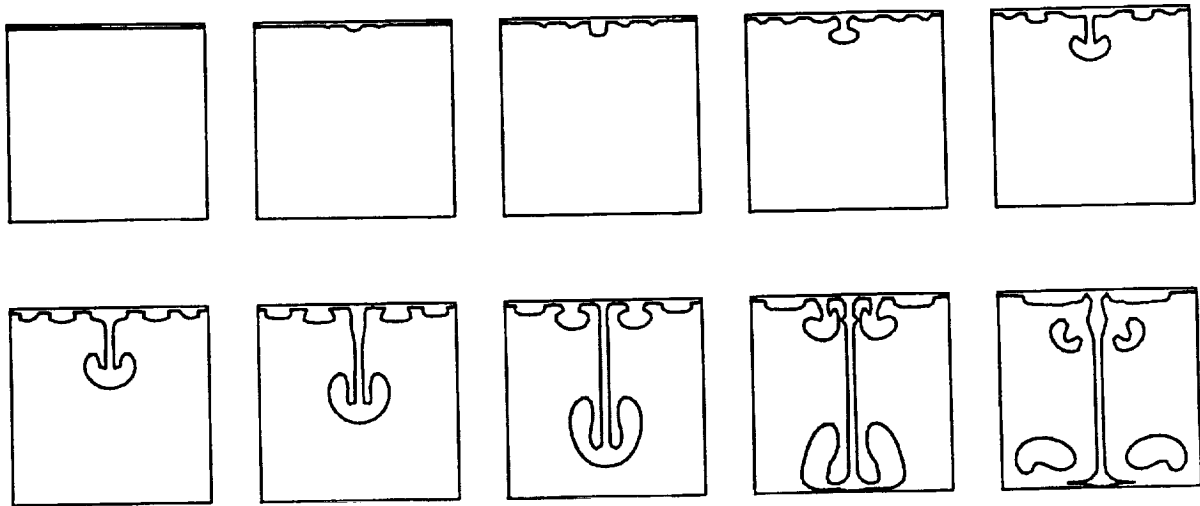


Figure 1.- An evolution of the density distribution for initial 4 seconds (a seed is given at a grid point only).

Figure 2 shows the density distributions for initial 8 seconds. The random numbers are used here for the initial distribution of the density. This evolution of the density will be also presented by the aid of a color animation video at the meeting. The color display makes it easy to understand the density distributions. In the beginning of the mixing, wavy motions appear as seen in figure 1 also. The amplitudes of the waves increases gradually, and some plumes of the heavier liquid grow. Two dominant plumes can be seen in the early stage of mixing. Finally, these mushroom-shaped plumes join into a vortical flow. After the dominant flow forms, following plumes are caught into the vortex one after another. The vortical flow develops further and the heavier part sinks downward. This feature of mixing is very similar to the observation in the experiment (ref. 7). It should be noted here that the position of the dominant plumes and the general behavior of mixing are not strongly affected by the artificial initial distribution of density.

Figure 3 shows an evolution of the density profiles in the vertical direction. The density is averaged in the horizontal direction. It is obvious that the front of the plumes moves downward. The density changes steeply at the front. This tells that the convective mass transport is dominant compared with the diffusive transport there. The distribution is basically plateau shaped, though some unevenness is there. The heavier liquid seems to be mixed well within the vortex.

Figure 4 shows the position of the front of plumes as a function of time. The position of the front is defined here as the point where scaled density is 0.1. The velocity increases with time, and reach some value. An interest thing is that the front pause for a moment after 6 seconds mixing. This behavior was observed in the experiment also. The plume will go down with intermittent pauses. The numerical result for the average plume velocity shows good agreement with the experimental result.

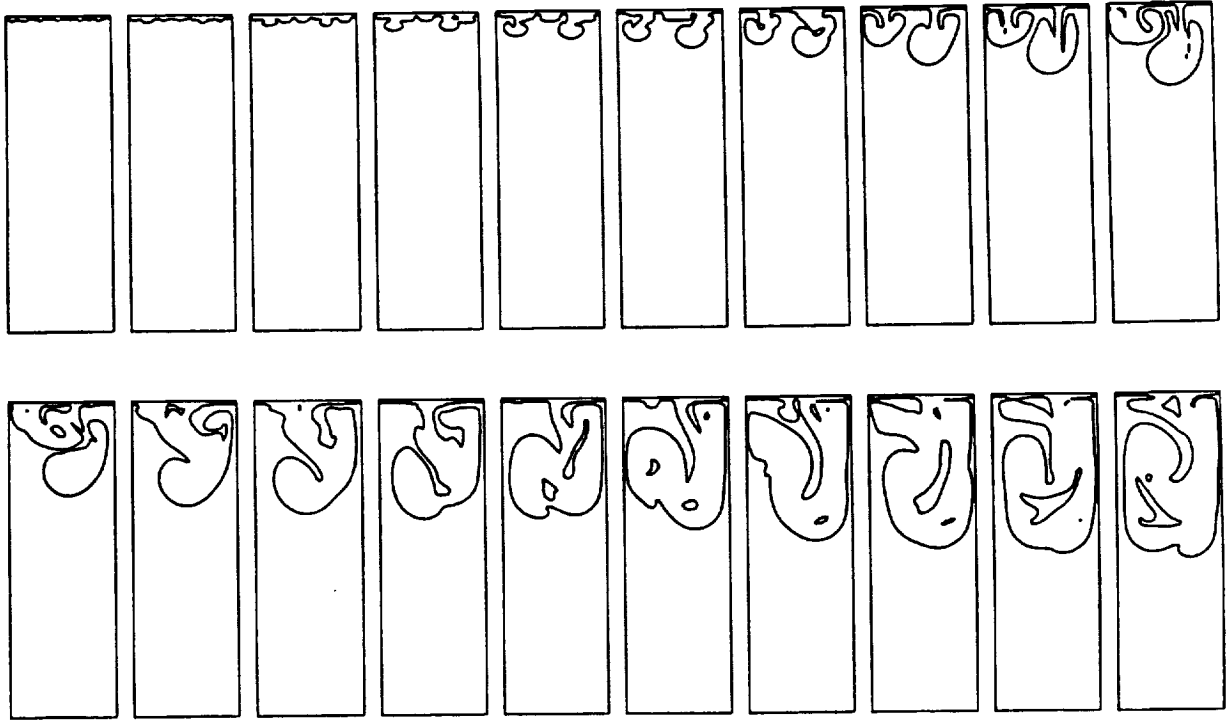


Figure 2.- An evolution of the density distribution for initial 8 seconds (a series of random numbers is used to specify initial distribution of density near free surface).

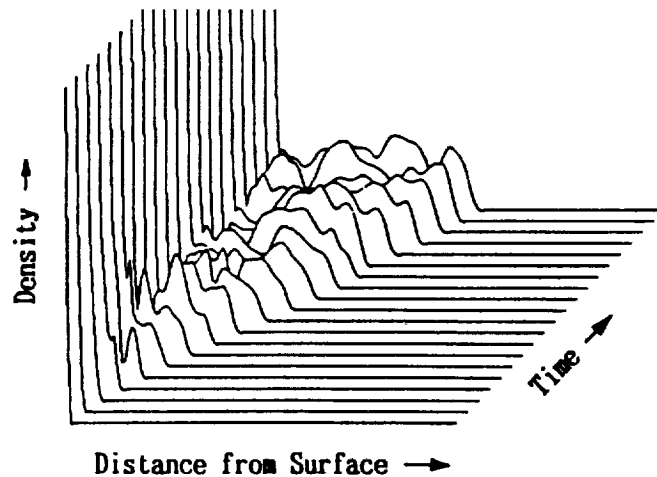


Figure 3.- A temporal change of the density profile.

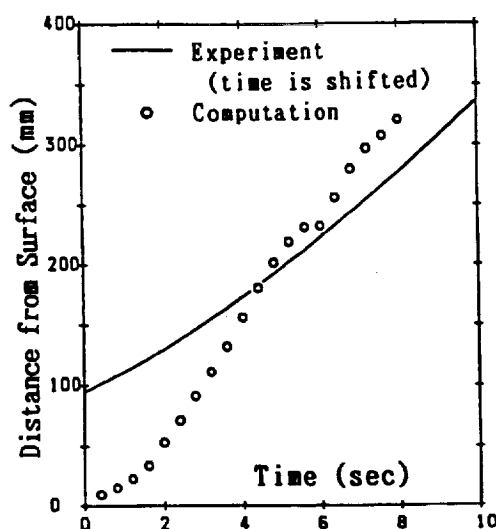


Figure 4.- Position of the front of the plumes as a function of time.

CONCLUSIONS

The mixing process of heavier liquid sprayed on the lighter layer has been analyzed numerically. The temporal change of the flow field and the density field are obtained. Realistically complex process is successfully predicted by using a series of random numbers for the initial condition. The present simulation catches the features, the momentary pause of the plumes' front, which is observed in the experiment. The numerical result for the plume velocity shows good agreement with the experimental result.

REFERENCES

1. Smith, K.A. et al.: Mixing and roll-over in LNG storage tanks. *Advances in Cryogenic Engineering*, vol.20, 1975, pp. 124-133.
2. Kusakari, K. et al.: Prevention of stratification of stock in LNG tank. IHI technical review, in Japanese, vol.22, no.5, 1975, 6p.
3. Oguz, H.N. et al.: Bubble entrainment by the impact of drops on liquid surfaces. *J. Fluid Mech.*, vol.219, 1990, pp. 143-179.
4. Hsiao, M. et al.: The critical wave number for vortex and jet formation for drops impinging on a liquid pool. *Phys. Fluids*, vol.31, no.12, 1988.
5. Kawamura, K. et al.: Direct simulation of a turbulent inner flow by finite-difference method. AIAA-85-0376, 1985, 10p.
6. Chandrasekhar, K.: *Hydrodynamic and hydromagnetic stability*. Chap.10, 1961.
7. Uchida, H. and Arai, T.: Experimental study on mixing of sprayed liquid in an LNG storage tank. being prepared.

ANALYSIS OF A SPACE EMERGENCY AMMONIA DUMP USING THE FLOW-NET TWO-PHASE FLOW PROGRAM

J. Navickas
McDonnell Douglas Space Systems Company
Huntington Beach, California

W.C. Rivard
Mechanical Engineering Department
University of Maine

ABSTRACT

Venting of cryogenic and non-cryogenic fluids to a vacuum or a very low pressure will take place in many space-based systems that are currently being designed. This may cause liquid freezing either internally within the flow circuit or on external spacecraft surfaces. Typical ammonia flow circuits were investigated to determine the effect of the geometric configuration and initial temperature, pressure, and void fraction on the freezing characteristics of the system. The analysis was conducted also to investigate the ranges of applicability of the FLOW-NET program. It was shown that a typical system can be vented to very low liquid fractions before freezing occurs. However, very small restrictions in the flow circuit can hasten the inception of freezing. The FLOW-NET program provided solutions over broad ranges of system conditions, such as venting of an ammonia tank, initially completely filled with liquid, through a series of contracting and expanding line cross sections to near-vacuum conditions.

INTRODUCTION

Freezing of either a cryogenic or a non-cryogenic fluid during venting to a low pressure environment is a concern in the design of the space liquid storage systems. There are many situations where such venting may be necessary. For instance, the Space Station ammonia loop may have to be vented during an emergency, liquid delivered to orbit may have to be dumped during a shuttle emergency, or liquid may have to be transferred from a high-pressure supply tank to a tank at a low pressure. During the design phases of such orbital liquid systems the question is often asked whether most of the liquid can be vented prior to freezing, and in which specific locations in the flow circuit can such freezing occur.

The design complexity of the space fluid systems requires that adequate computational tools be available for the analysis of such systems. During the venting process a typical storage system will start out with a storage tank initially at a high pressure and a low void fraction. As liquid flows through the system, it increases in void fraction, decreases in temperature, and can result in a completely evaporated liquid or a single-phase vapor flow at the outlet to space. The analysis of such a system is rather demanding computationally and the methods of analysis are currently under development and have not reached a state of maturity where such problems can be solved routinely. Typical designs considered in the present analysis

were analyzed both to provide some insights into the freezing problem and to check out the applicability of the FLOW-NET program.

COMPUTATIONAL METHOD

The FLOW-NET program was used to conduct the computations. The initial version of the program, called SOLA-LOOP, was developed at the Los Alamos Scientific Laboratory (ref. 1). The program development continued at Flow Science, Inc. and at the present time the development continues at the University of Maine.

Conservation of mass, energy, and momentum equations solved by the program are given in reference 2. They are repeated here for the sake of completeness.

Conservation of mass

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u A}{A \partial x} = \frac{S_g + S_c}{V}$$

$$\frac{\partial \rho_g}{\partial t} + \frac{\partial A}{A \partial x} \left(\rho_g u + \frac{\rho_g \rho_l u_r}{\rho} \right) = J_c + J_v + \frac{S_g}{V}$$

$$\frac{\partial \rho_l}{\partial t} + \frac{\partial A}{A \partial x} \left(\rho_l u + \frac{\rho_l \rho_l u_r}{\rho} \right) = \frac{S_l}{V}$$

Conservation of momentum

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) + \frac{1}{\rho A} \frac{\partial}{\partial x} \left(A \rho_g \rho_l \frac{u_r^2}{\rho} \right) = - \frac{\partial P}{\rho \partial x} + g_x - \frac{1}{2} (f_w, f_L) u |u| + \dot{u}_p$$

$$\frac{\partial u_r}{\partial t} + \frac{\partial}{\partial x} \left\{ u_r \left[\frac{u + u_r (\rho_l - \rho_g)}{2\rho} \right] \right\} = \left(\frac{1}{\rho_l} - \frac{1}{\rho_g} \right) \frac{\partial P}{\partial x} - \frac{K_d \rho u_r}{\rho_g \rho_l} - u_r \left(\frac{J_c}{\rho_g} + \frac{J_v}{\rho_l} \right)$$

Conservation of energy

$$\frac{\partial}{\partial t} (\rho_g E_g + \rho_l E_l) + \frac{\partial A}{A \partial x} \left\{ u_g [\rho_g E_g + P \theta] + u_l [\rho_l E_l + P (1-\theta)] \right\} = \frac{Q_m + S_i H_i + S_v H_v + S_l H_l}{V}$$

In these equations,

- A = cross sectional area
- E_g = specific total gas energy
- E_l = specific total liquid energy
- f_L = area change loss coefficient
- f_w = friction loss coefficient
- g_x = body acceleration
- H_i = noncondensable gas enthalpy
- H_l = liquid enthalpy
- H_v = vapor enthalpy
- J_c = rate of condensation

J_e	= rate of evaporation
K_d	= liquid-gas momentum exchange coefficient
P	= pressure
t	= time
S_g	= external gas source
S_i	= external noncondensable gas source
S_e	= external liquid source
S_v	= $S_g - S_i$
u	= mixture velocity
u_r	= relative velocity between liquid and vapor
\dot{u}_p	= acceleration due to externally applied force
V	= mixture volume
x	= distance
θ	= void fraction
ρ	= mixture density
ρ_g	= vapor density
ρ_i	= noncondensable gas density

06/26/92

The equations of motion as presented here are equivalent to those in other two-phase flow programs such as ATHENA (ref. 3), RELAP5 (ref. 4), and TRAC (ref. 5). In these equations K_d is a function that describes the momentum exchange between the liquid and the gas phases. Large differences in liquid-gas velocities are associated with a small value of K_d . Conversely, small differences in liquid-gas velocities are associated with a large value of K_d . Initially the FLOW-NET program was formulated assuming relatively small velocity differences. This permitted the elimination of terms underlined in the momentum equation. These terms have been included in a recent program modification (ref. 6). Although the capability to solve problems with large velocity differences exists, such problems can be solved only when K_d is known, which, in most cases, has to be determined experimentally. Fortunately, there are classes of problems where K_d could be safely assumed to be large. System venting problems considered here can be assumed to have small relative velocities between phases. In such a case, the continuously decreasing system pressure will cause nucleation and continuous vapor generation. Such nucleation usually occurs at solid surfaces, thus breaking up any tendency to separate the phases into a low-velocity liquid phase attached to the solid surfaces and a high-velocity vapor core. This is illustrated in Figure 1.

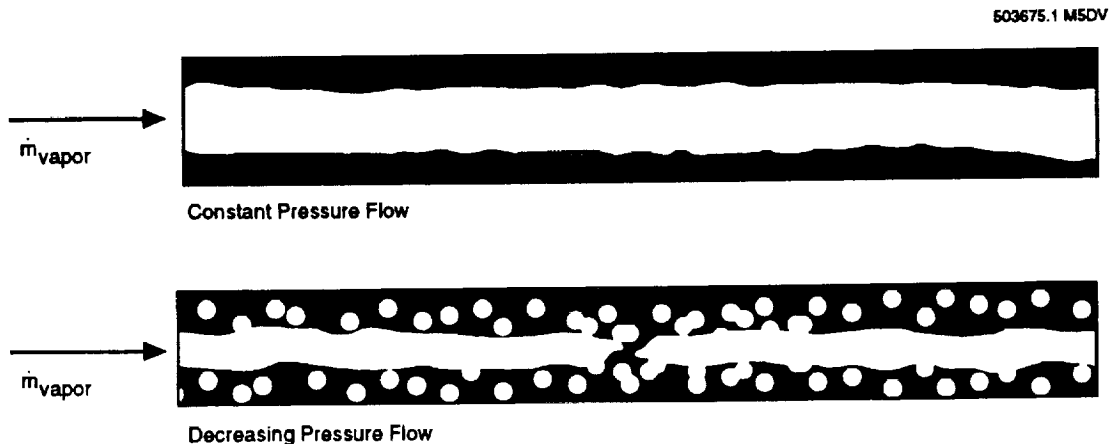


Figure 1. Effect of pressure decrease on flow configuration

NUMERICAL SOLUTIONS

The venting analyses presented in this paper were conducted as part of the Two-Phase Integrated Thermal System (TPITS) shuttle experiment designed to evaluate the Space Station two-phase ammonia thermal control system. Although the analyses were performed to evaluate a specific system, results are applicable to other similar systems and show significant trends and design conditions to be avoided.

Two specific analyses were conducted. In the first one a typical ammonia tank was vented to space. A long-duration run was made to determine the point where freezing is likely to occur. In the second case short runs were conducted to determine the effects of flow restrictions in the vent line and initial tank thermodynamic conditions.

1. Venting of an Ammonia Tank

The computational model considered is shown in Figure 2. The supply tank initial conditions were $P = 67.0$ psia, $T = 35^\circ\text{F}$. The outside boundary pressure was kept at 31.16 psia for 10 seconds, then allowed to decrease to 1.24 psia in 300 seconds. The pressure was not decreased any further to avoid temperature decrease below freezing, a condition that has no physical meaning, because the program can consider only liquid-vapor mixtures with no solid phase. The saturation temperature corresponding to the 1.24 psia pressure is -100°F , slightly above the -107.86°F freezing temperature.

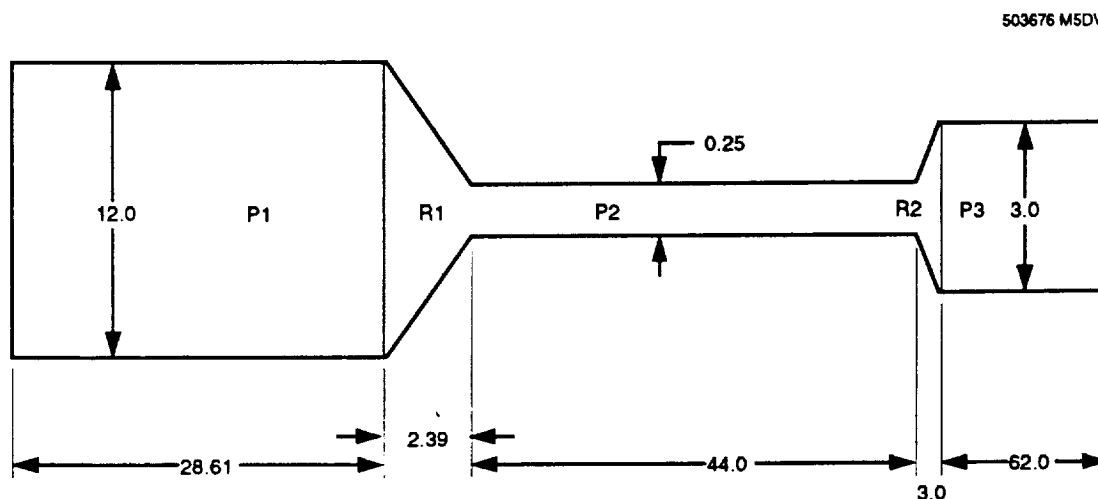


Figure 2. Ammonia dump computational model

The computational model consists of pipe P1, representing the supply tank, a reducer R1 between the supply tank and the vent line P2. Expander R2 and pipe P3 are approximations of space conditions. There are no reliable methods available that could be used to solve liquid-vapor plume problems. Only crude approximations of external conditions can be made, as were done in the present analysis. To approximate the external conditions, the flow was allowed to expand from a 0.25 in. diameter line to a 3.0 in. diameter line. Results of such analysis can give a qualitative indication of possible ice formation outside of the exit plane.

The supply tank void fraction, mixture pressure, and temperature histories are shown in Figures 3, 4, and 5, respectively. The aim of a liquid dump system is to vent as much of the liquid as possible without freezing the liquid. As shown in Figure 3, the tank approaches a void fraction of 1.0 at approximately 350 seconds. At this point the temperature, as shown in Figure 5, is well above the freezing temperature. It can be concluded that the tank can be vented without freezing. At 350 seconds the tank vent line exit temperature is well above the freezing point, as shown in Figure 6. It can also be concluded that the liquid in the line will not freeze.

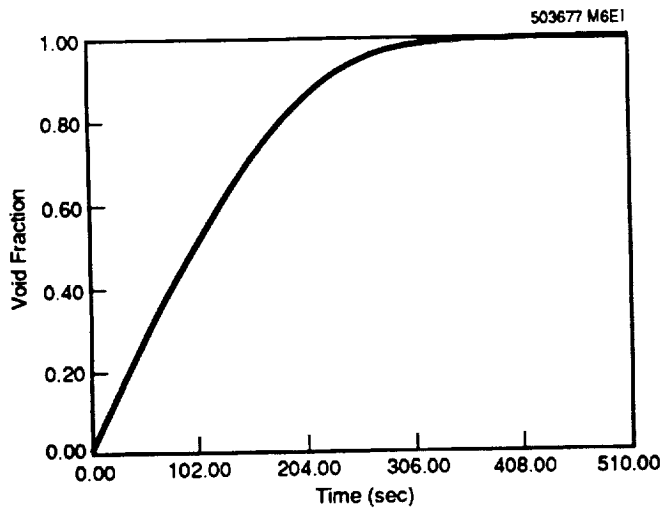


Figure 3. Storage tank void fraction history

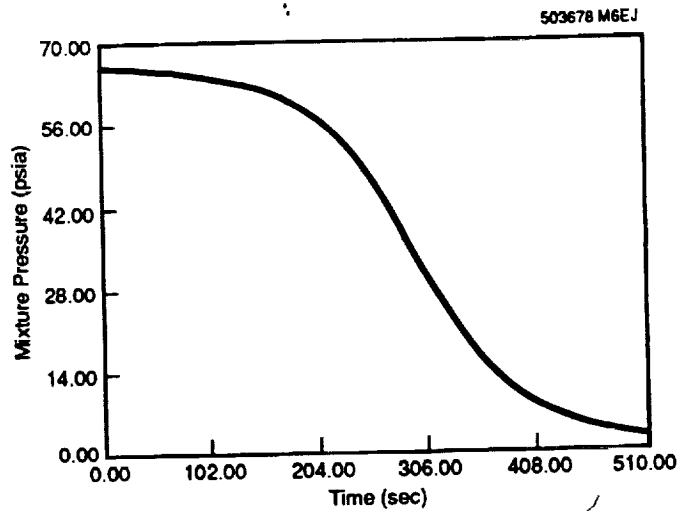


Figure 4. Storage tank pressure history

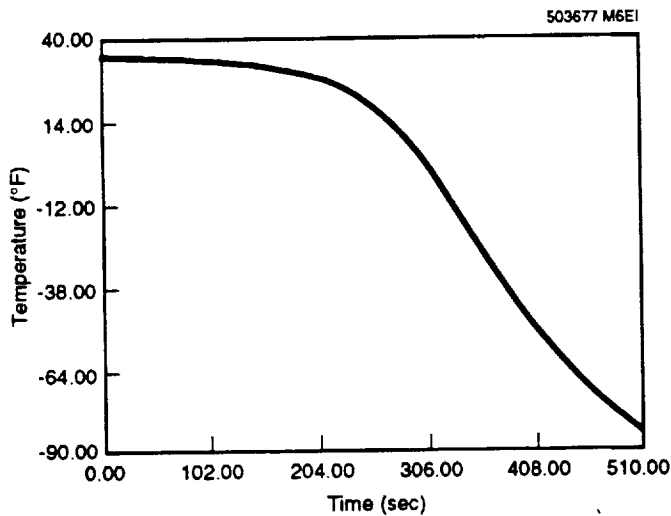


Figure 5. Storage tank temperature history

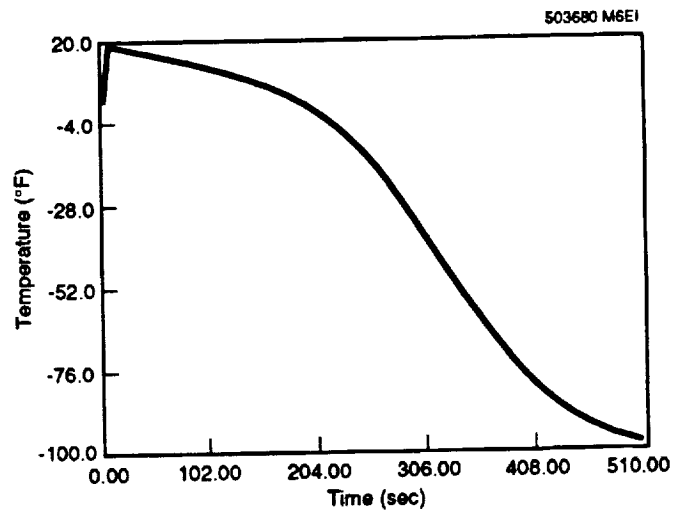


Figure 6. Mixture temperature at vent pipe exit

Temperature distribution along the length of the circuit is shown in Figure 7. Results show that the temperature at the exit drops substantially between 200 and 300 seconds, getting close to freezing between 300 and 400 seconds. It can, therefore, be concluded that conditions outside of the vent exit could cause some freezing. The total mass that could freeze is small because the void fraction during this time period is large as shown in Figure 8.

2. Effect of Geometry and Initial Conditions on the Vent System Performance

A flow circuit with contracting-expanding cross section was constructed to get some understanding of the effect of the flow circuit geometry and initial conditions on the freezing potential. The flow circuit geometry is shown in Figure 9; the conditions analyzed are given in Table I. Circuit exit boundary conditions are shown in Figure 10. In this particular case the exit pressure was reduced to 0.2 psia, thus creating a potential to reach temperatures below freezing. However, it should be realized that temperatures below freezing have no physical meaning. The solution gives an indication that a freezing condition is approaching but gives no quantitative answers.

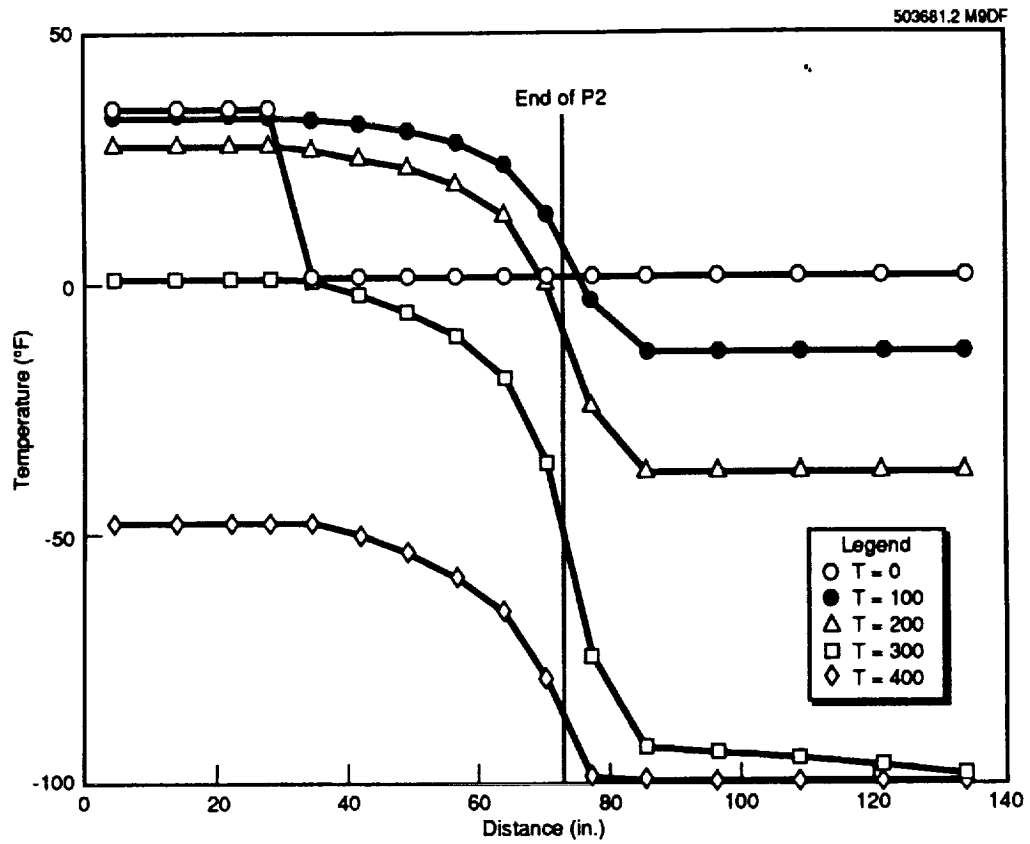


Figure 7. Temperature distribution along the length of the flow circuit

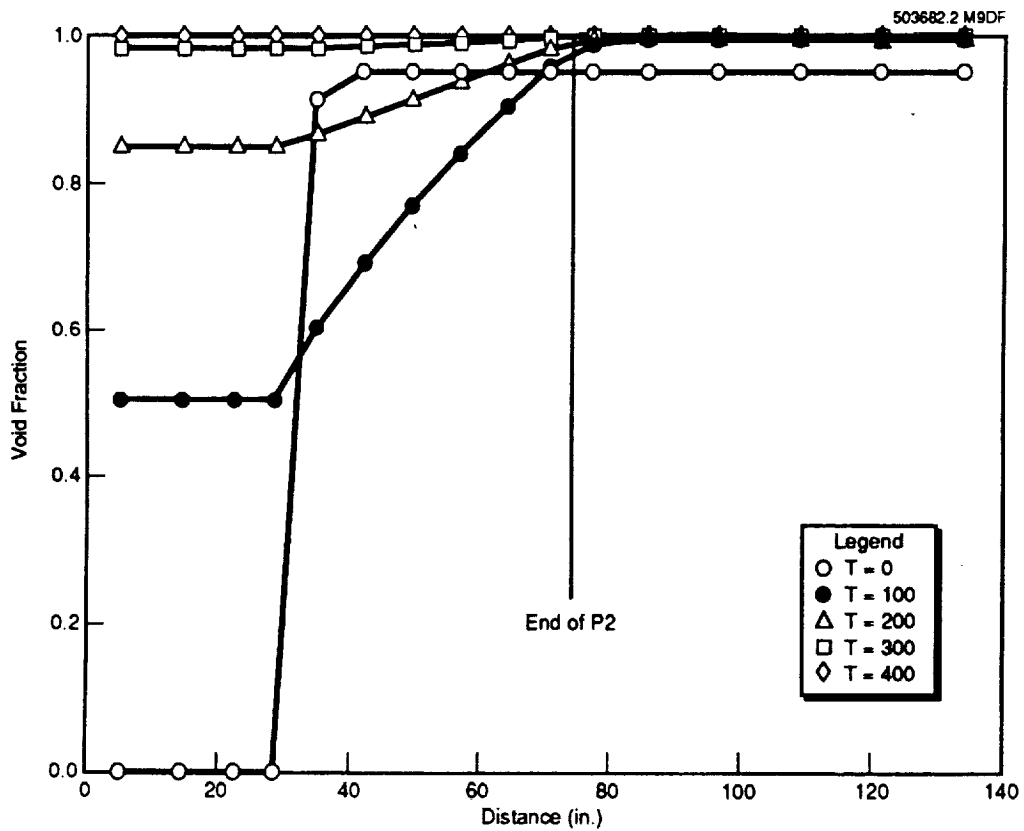
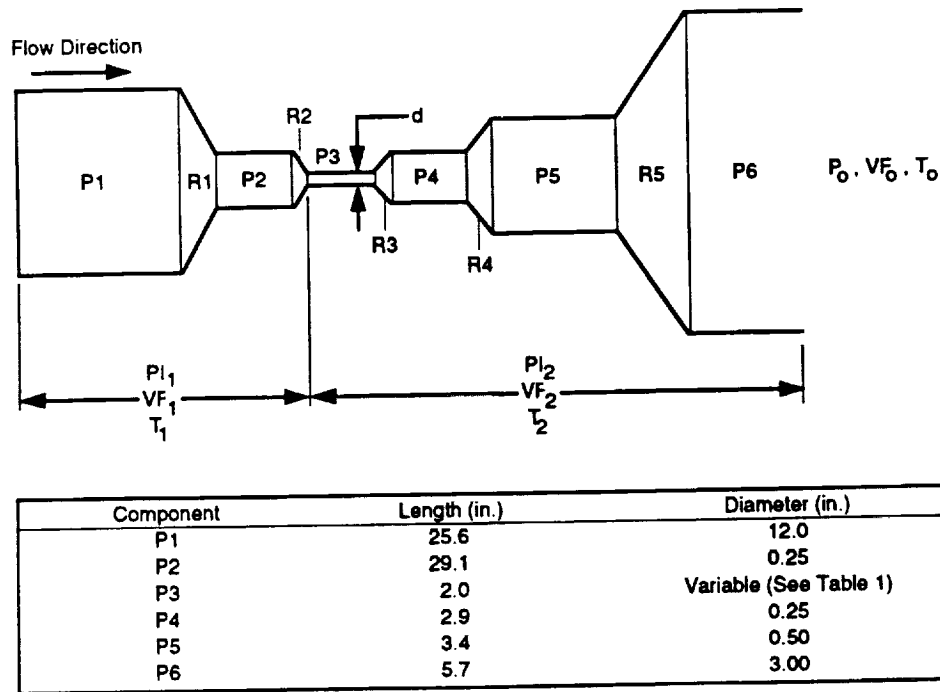


Figure 8. Void fraction distribution along the length of the flow circuit



P_i , VF_i , T_i – initial pressure, void fraction, and temperature (Table 1)

P_o – boundary pressure (Figure 10)

Figure 9. Flow circuit geometry

TABLE I. INITIAL CONDITIONS AND THE THROAT DIAMETER

	Case 1	Case 2	Case 3	Case 4
PI_1 (psi)	67.0	8.0	8.0	8.0
VF_1	0.001	0.99	0.99	1.0
T_1 (°F)	35.0	-50.0	-50.0	-50.0
PI_2 (psi)	32.0	8.0	8.0	8.0
VF_2	1.0	1.0	1.0	1.0
T_2 (°F)	2.0	-50.0	-50.0	-50.0
d (in)	0.056	0.056	0.20	0.056

prof-paper: 92h0533 t1 06/25/92

To conserve computer time, the four cases were run for 2.0 seconds. Temperature histories for the four cases are shown in Figure 11. Plots start at 0.5 seconds to give a better resolution of the results. Case 1, which is similar to the first long-duration case described in this paper, shows a very gradually decreasing temperature, well above the freezing point. Case 2, which has a low supply tank pressure and a high void fraction (a condition that could be encountered partially into the vent cycle) shows a rapidly decreasing temperature, reaching a point well below freezing. Case 3, similar to Case 2, but with a larger restriction diameter, shows much higher temperature. This gives some quantitative evidence to an intuitively obvious fact that severe restrictions can hasten the formation of ice within the flow circuit. It is, therefore, prudent to avoid such restrictions whenever possible. Case 4 shows the temperature response of a pure vapor, to show the difference in response to a two-phase medium.

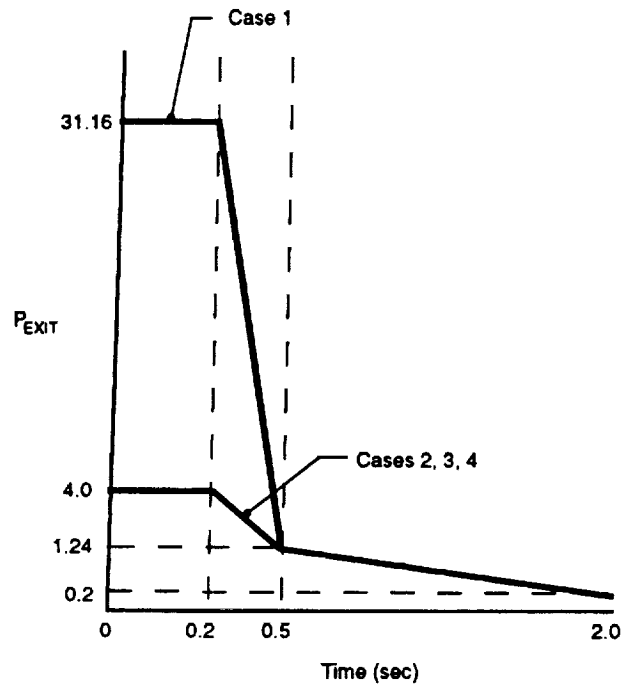


Figure 10. Exit pressure histories

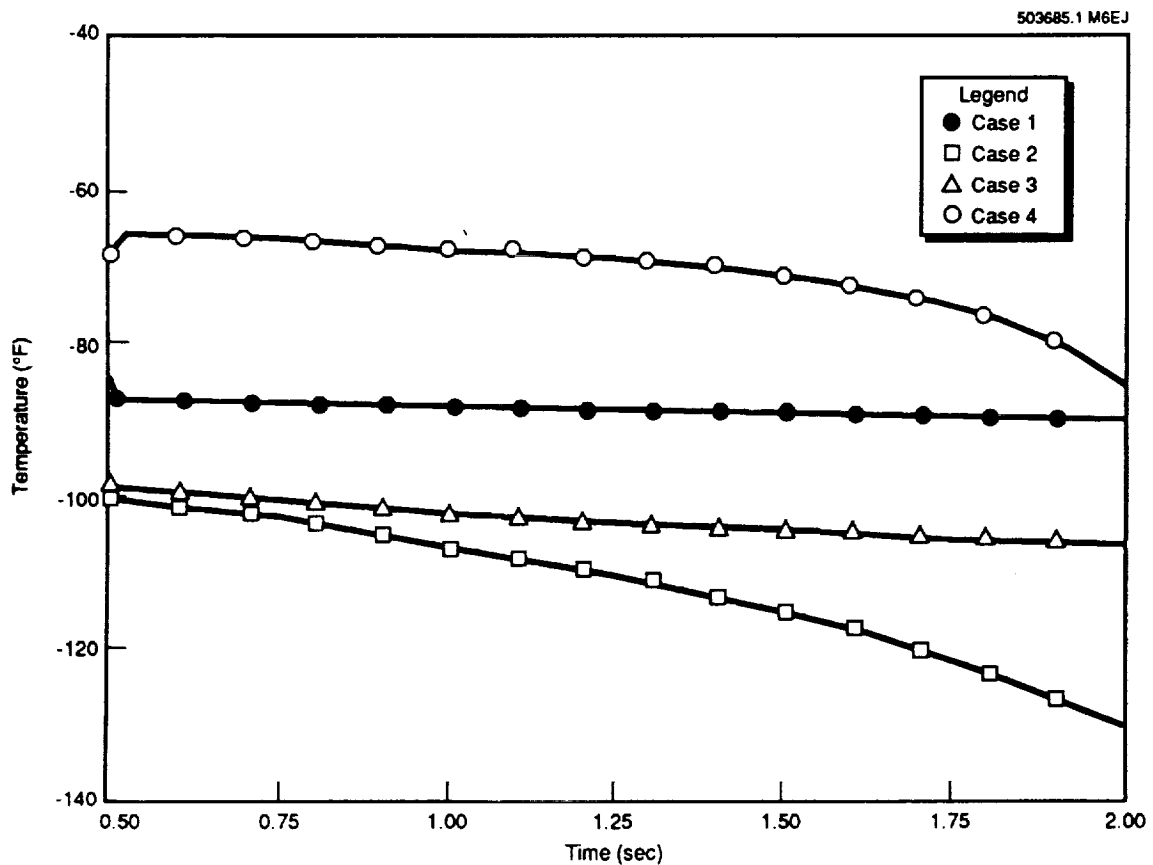


Figure 11. Pipe P5 exit temperature history

CONCLUSIONS

The ammonia dump analysis results indicate that ammonia can be readily vented overboard either in an emergency situation or under normal operating conditions with little danger of freezing. For a typical flow circuit considered in the analysis, practically all of the ammonia could be vented with temperatures remaining well above freezing. Freezing potential developed outside of the vent nozzle toward the end of the dump operation, when the liquid quantity in the system is low. Restriction in the flow circuit can greatly enhance the possibility of freezing and should be avoided. When such restrictions cannot be avoided, they should be included in the computational model.

No difficulties were encountered using the FLOW-NET program. The cases analyzed are rather difficult cases computationally, since the flow starts out as essentially a pure liquid in the supply tank, undergoes a phase change in the flow circuit, then expands into essentially space environment. All results were stable and the solutions well-behaved.

REFERENCES

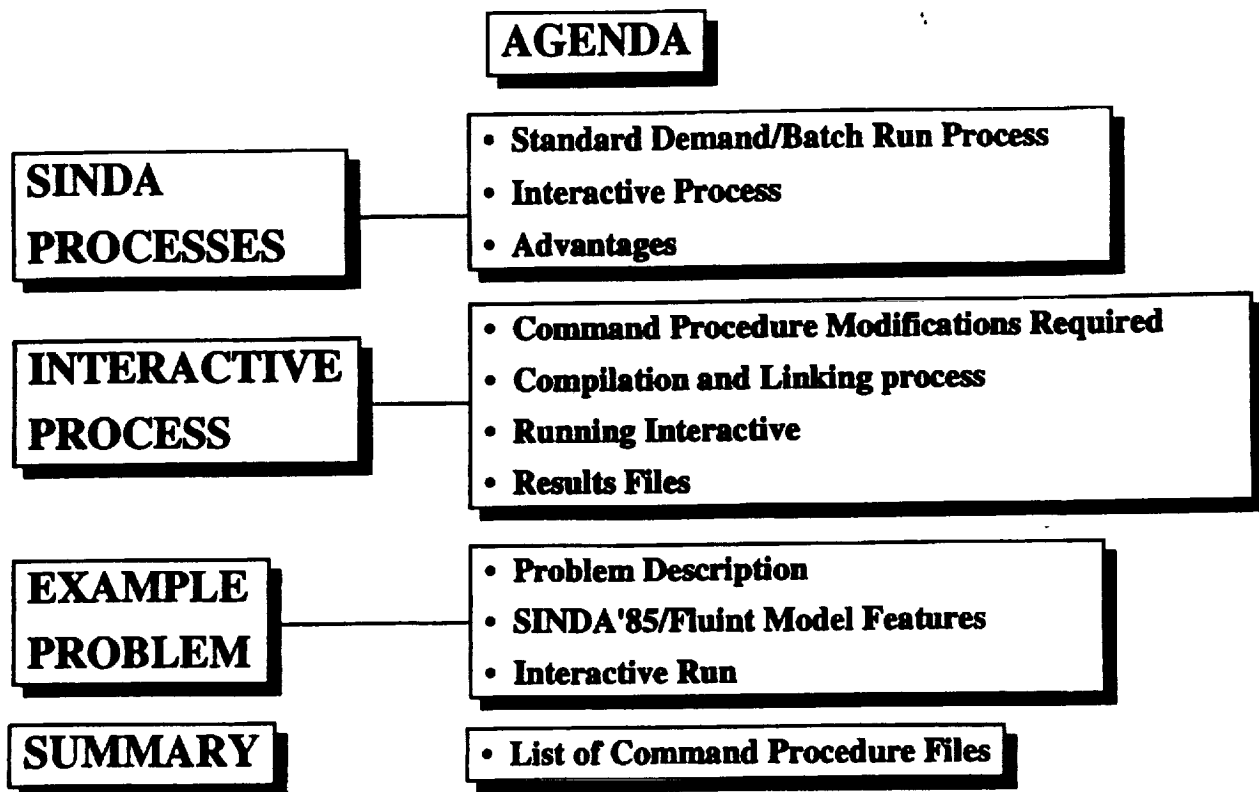
1. Hirt, C.M.; Oliphant, T.A.; Rivard, W.C.; Romero, N.C.; Torrey, M.D.: SOLA-LOOP: A Nonequilibrium, Drift-Flux Code for Two-Phase Flow in Networks. Los Alamos Scientific Laboratory report NUREG/CR-0626, June 1979.
2. Rivard, W.C.; Navickas, J.: Analysis of CO₂ Fire Extinguisher Discharge Using the FLOW-NET Program. Presented at the 22nd International Conference on Environmental Systems, Seattle, Washington, July 13-16, 1992.
3. Carlson, K.E.; Roth, P.A.; and Ransom, V.H.: ATHENA Code Manual Volume 1: Code Structure, System Models, and Solution Methods. Idaho National Engineering Laboratory report EGG-RTH-7397, September 1986.
4. Ransom, V.H., et al.: RELAP5/MOD1 Code Manual Volume 1: System Models and Numerical Methods. Idaho National Engineering Laboratory report EGG-2070, NUREG/CR-1826, March 1982.
5. Thermal Reactor Safety Group. TRAC-P1: An Advanced Best Estimate Computer Program for PWR LOCA Analysis I. Methods, Models, User Information, and Programming Details. Los Alamos Scientific Laboratory report LA-7279-MS, NUREG/CR-0063, May 1978.
6. Rivard, W.C.: A Two-Fluid Model for the FLOW-NET Computer Program. Final report, University of Maine, work conducted for the McDonnell Douglas Space Systems Company, September 14, 1991.

RUNNING SINDA '85/FLUNT INTERACTIVE ON THE VAX

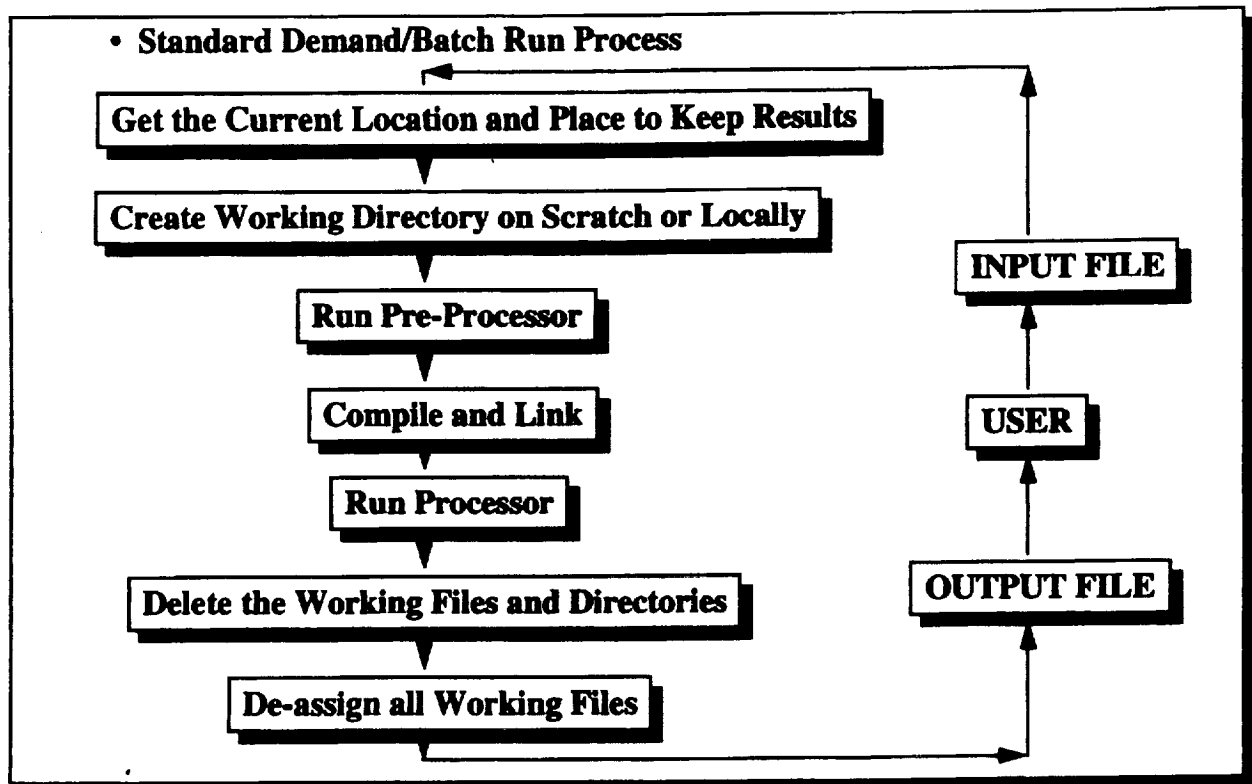
Boris Simmonds
Sverdrup Technology MSFC Group
Huntsville, Alabama

ABSTRACT

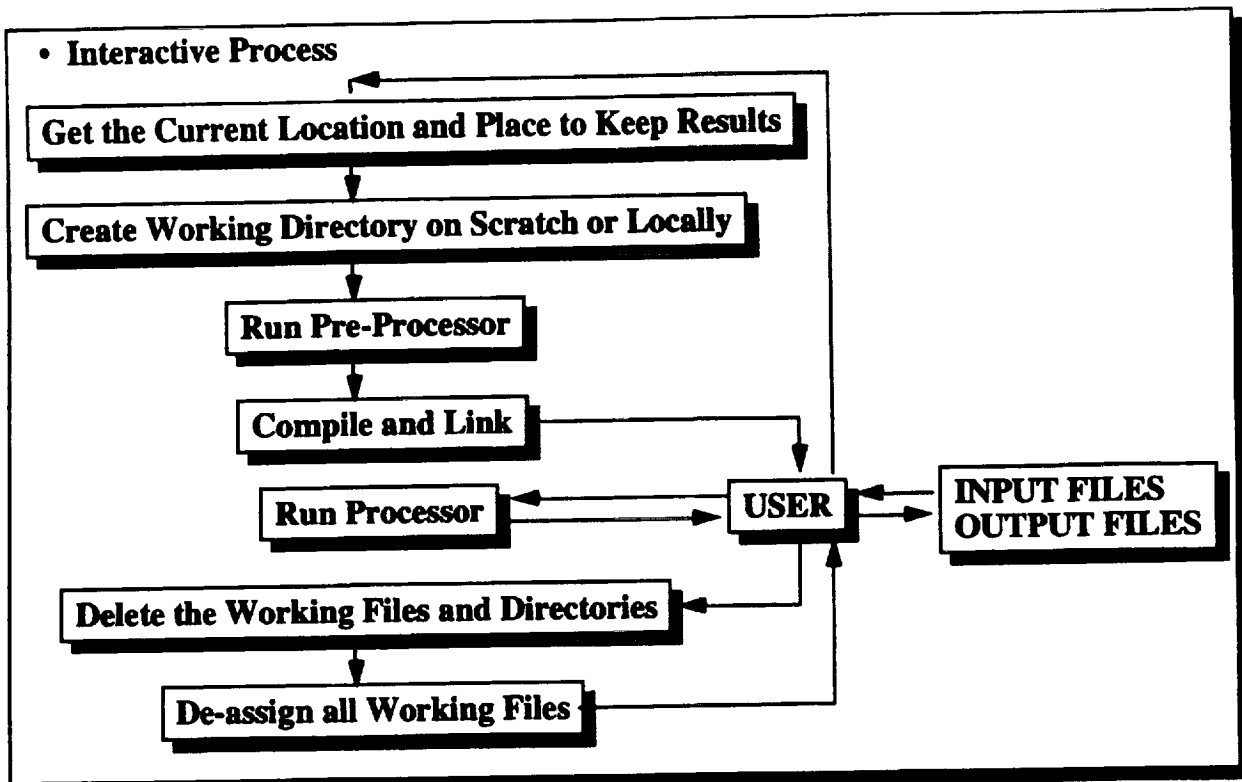
Computer software as engineering tools are typically run in three modes: Batch, Demand and Interactive. The first two are the most popular in the SINDA world. The third one is not so popular, due probably to the users inaccessibility to the command procedure files for running SINDA '85, or lack of familiarity with the SINDA '85 execution processes (pre-processor, processor, compilation, linking, execution and all of the file assignment, creation, deletions and de-assignments). Interactive is the mode that makes thermal analysis with SINDA '85 a real-time design tool. This paper explains a command procedure sufficient (the minimum modifications required in an existing demand command procedure) to run SINDA '85 on the VAX in an interactive mode. To exercise the procedure a sample problem is presented exemplifying the mode, plus additional programming capabilities available in SINDA '85. Following the same guidelines the process can be extended to other SINDA '85 residence computer platforms.



SINDA PROCESS



SINDA PROCESS



INTERACTIVE PROCESS

• Minimum Command Procedure Modifications Required

- In the Standard ASTA.COM File Hold the Run-Process and the File-Deletion-Process by Commenting the Following Two Lines:

```
$ RUN 'FNAME  
$ @AST:DELWORK
```

- Recommend You Create a New File such as ASTA_SAVE.COM.
- Define a Symbol such as SINDA85_SAVE:= = @ASTA_SAVE in Your THERMAL Set-Up or LOGIN.COM Files.

INTERACTIVE PROCESS

- **Compilation and Linking Process**

- **Run the Preprocessor With Input File XXX.INP:**

\$>SINDA85_SAVE XXX.INP

- **If Errors are found, files XXX.OPP or XXX.LIS located in the Same Directory of XXX.INP Will contain any Pre-Processor (SINDA85/Fluint) or Compilation (Fortran) error messages.**

- **If No Errors, You will find Yourself within the ZZZZZ.DIR Scratch Directory. Among All of the XXX.DAT files is the XXX.EXE Executable ready to Run.**

- **Transparent to the User, the VAX System Has also Assign a Number of Working Files (Just like your ASTA_SAVE.COM File Requested That Will Remain Assign Until they Are De-assigned, or Until You Logout. Should You Logout, These Assignments Need to be Made Before the XXX.EXE can be Run.**

INTERACTIVE PROCESS

- **Running Interactive:**

- **To Run Just Enter:**

\$>RUN XXX

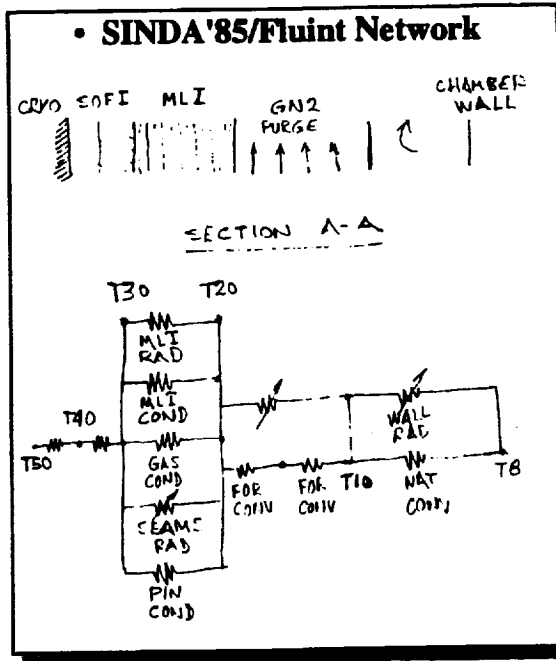
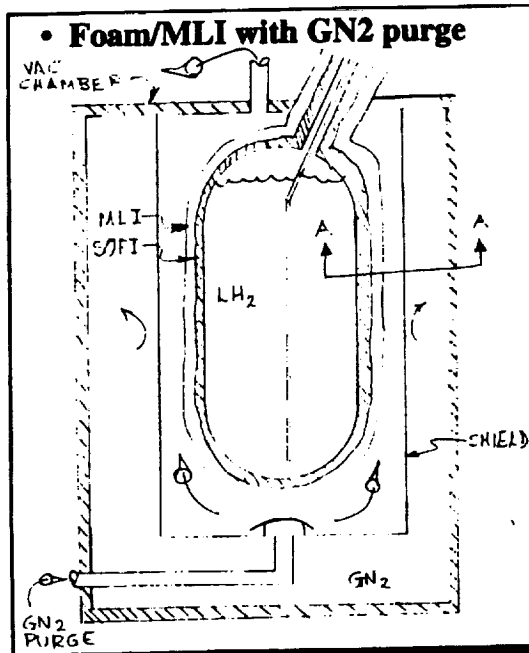
- **All of the Lines Programmed in the HEADER OPERATIONS DATA Block of the SINDA '85/Fluint Model Will Begin Execution.**

- **Result Files:**

- **Result Files (XXX.OUT, XXX.US1, XXX.RSO, Etc) Will be Created in the Same Location as the XXX.INP Model.**

EXAMPLE PROBLEM

• Problem Description



PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP

```
C THIN SUBMODEL: DESIGNED TO MODEL THE THIN INSULATED
C SECTION OF THE FOAM/MLI TEST ARTICLE IN GROUND PHASE CONDITIONS.
C
C PURGE FLUID SUBMODEL: DESIGNED TO MODEL THE GN2 PURGE BETWEEN THE
C MLI AND THE SHIELD. THIS SUBMODEL GENERATES THE CONVECTION
C BETWEEN THE SHIELD AND THE THIN MLI INTERFACE.
```

HEADER OPTIONS DATA

```
TITLE FOAM/MLI GROUND PHASE
MODEL = TEST
OUTPUT = FMLIOUT
USER1 = FMLI.US1
```

C FLUID DESCRIPTION FOR LN2 PURGE GAS

```
HEADER FPROP DATA,8728,SI,0.0
```

```
C MOST COMPLETE N2 GAS (NEAR 1 ATM.)
```

```
C VALUES BELOW 77.36K ARE FOR VAPOR
```

```
C RGAS = 8314.34/28.01
```

```
AT,V, 65.0,4.40E-6
```

```
77.36,5.44E-6, 80.0,5.59E-6, 85.0,5.9E-6, 90.0,6.22E-6
95.0,6.54E-6, 100.0,6.87E-6, 105.0,7.19E-6, 110.0,7.52E-6
115.0,7.83E-6, 120.0,8.15E-6, 125.0,8.0E-6, 126.2,8.65E-6
130.0,8.78E-6, 140.0,9.4E-6, 180.0,11.8E-6, 200.0,12.9E-6,
220.0,13.9E-6, 240.0,15.0E-6, 280.0,16.0E-6, 280.0,16.9E-6,
300.0,17.9E-6, 340.0,19.7E-6, 440.0,23.7E-6, 460.0,24.4E-6,
480.0,25.2E-6, 500.0,25.9E-6
```

```
AT,K, 65.0,6.1E-3, 75.0,7.1E-3, 77.36,7.4E-3, 80.0,7.6E-3,
85.0,8.0E-3, 90.0,8.5E-3, 95.0,8.9E-3, 100.0,9.4E-3,
105.0,9.8E-3, 110.0,10.3E-3, 115.0,10.7E-3, 125.0,11.7E-3,
130.0,12.1E-3, 150.0,13.9E-3, 160.0,14.7E-3, 180.0,16.5E-3,
200.0,18.3E-3, 220.0,19.9E-3, 240.0,21.6E-3, 300.0,26.0E-3
320.0,27.4E-3, 340.0,28.7E-3, 380.0,31.3E-3, 400.0,32.5E-3,
480.0,37.5E-3, 500.0,38.6E-3
```

```
AT,CP, 65.0,1.039E3, 320.0,1.039E3, 380.0,1.042E3, 460.0,1.050E3
500.0,1.056E3
```

```
C-----
HEADER CONTROL DATA,GLOBAL
```

```
UID = ENG PATMOS = -14.7
ABSZRO = 0.0
SIGMA = 1.0
NLOOPS = 500
```

PROBLEM DESCRIPTION

• SINDA '85/Fluint Model FMLI.INP (cont)

HEADER USER DATA, GLOBAL

PI = 3.1416
VALUE = 0.0
ICASE = 1

C*****
C TO CHANGE MODEL CONFIGURATION, MODIFY THE FOLLOWING VARIABLES:
C*****

C CRYO PHYSICAL PROPERTIES:

QEVAP = 191.9 \$ HEAT OF VAPORIZATION OF CRYO (BTU/LBM)
TCRYO = 37.0 \$ CRYO TANK TEMP (DEG R)

C BOUNDARY CONDITIONS:

TWALL = 530.0 \$ CHAMBER WALL (DEG R)
TSHIELD = 520.0 \$ ALUMINUM SHIELD TEMP (DEG R)
GN2PT = 530.0 \$ CHAMBER GN2 PURGE GAS TEMP (DEG R)

C MLI PROPERTIES:

AMLITN = 45.809 \$ THIN MLI SURFACE AREA
DMLITN = 50.0 \$ MLI DENSITY (LAYERS/IN)
XLAYTN = 17.0 \$ NUMBER OF MLI LAYERS ON THIN (INC OUTER/INNER)
EMLIH = 0.05 \$ MLI HEMISPHERICAL EMISIVITY
EMLJO = 0.1 \$ MLI OUTER LAYER EMISIVITY PDIA = 1.32/12.0 \$ LEXAN PIN
DIA (FT)
DPIN = 1.0 \$ PIN DENSITY (#/SQFT)
FMSTN = 1.0 \$ THIN MLI-SHIELD VIEW FACTOR
SEAMLTN = 36.061 \$ THIN MLI SEAM LENGTH (FT)
SEAMWTN = 0.125/12. \$ THIN MLI SEAM WIDTH (FT)

C SOFI PROPERTIES:

ASOFITN = 43.986 \$ THIN SOFI AREA (FT2)
SOFITN = 0.45/12. \$ THIN SOFI THICKNESS (FT)

C WALL PROPERTIES:

EWALL = 0.8 \$ EMISSIVITY OF VAC WALL CHAMBER

C SHIELD PROPERTIES:

ASHTN = 62.03 \$ THIN SHIELD AREA (FT2)
ESHIELD = 0.9 \$ EMISSIVITY OF AL SHIELD
FSW = 1.0 \$ SHIELD-WALL VIEW FACTOR
CONVSW = 1.0 \$ GN2 CONV BET SHIELD AND WALL (BTU/HR-FT2-F)

C GN2 PURGE FLUINT NETWORK

GN2MLIT = 530.0 \$ MLI GN2 PURGE GAS TEMP (DEG R)
GN2PFR = 10.0 \$ MLI GN2 PURGE FLOWRATE (LBS/MIN)
HTCTN = 0.0 \$ THIN GN2 PURGE H(BTU/HR-FT2-F) (OUTPUT)
RENTN = 0.0 \$ THIN GN2 REYNOLD'S NO. (OUTPUT)

C USED FOR OUTPUT ONLY XMLITN = 0.0 \$ THIN MLI THICKNESS
VPURTN = 0.0 \$ GN2 VEL BET SHIELD AND THIN MLI (FT/SEC)

C OTHER MODIFIABLE INPUTS

STEF = 0.1714E-8 \$ STEFAN-BOLTZMANN (BTU/HR-FT2-R4)

C*****

C END OF MODIFICATIONS

C*****

PROBLEM DESCRIPTION

• SINDA '85/Fluint Model FMLI.INP (cont)

C THIN SUBMODEL:

C

C MAJOR ASSUMPTIONS

C 1. TANK OUTER SURFACE IS CONSTANT (DEG RANKINE)

C 2. CHAMBER WALL TEMPERATURE IS CONSTANT (DEG RANKINE)

C 3. AVERAGE SOFI THICKNESS IS 0.45 IN. (BASED ON THICKNESS MAP)

C 4. MLI DENSITY - 50 LAYERS/MIN., 15 LAYERS MLI PLUS TWO MYLAR COVERS

C MLI THICKNESS = $17/50 = 0.34$ IN.

C 5. MLI HEMISPHERICAL EMISSIVITY = .05

C 6. EMISSIVITY OF MLI OUTER SURFACE = .1, EMISSIVITY OF SHIELD = .9

C EMISSIVITY OF VACUUM CHAMBER = 0.8.

C 7. LEXAN PINS, DIAMETER = 1/8 IN., DENSITY = 1 PER FT² MLI

HEADER USER DATA, THIN

101=0.

102=0.

103=0.

201=0.

301=0.

401=0.

501=0.

502=0.

503=0.

504=0.

C

888=0.

999=0.

HEADER NODE DATA, THIN

10, 520.0, -1.0 \$ ALUMINUM SHIELD

20, 450.0, -1.0 \$ MLI SURFACE

30, 360.0, -1.0 \$ SOFI SURFACE

40, 200.0, -1.0 \$ SOFI MIDPOINT

-8, 530.0, 0.0 \$ CHAMBER WALL

-9, 530.0, 0.0 \$ GN2 PURGE GAS

-50, 37.0, -1.0 \$ CRYO

HEADER CONDUCTOR DATA, THIN

C CALCULATION FOR HEAT LEAK COMPONENTS

C G(mil-shield) (100) = $A \cdot F_v \cdot F_e \cdot STEF$

C G(mil cond) (101) = (CALCULATED IN VARIABLES 1 USING EMPIRICAL FORMULA)

C G(mil rad) (102) = (CALCULATED IN VARIABLES 1 USING EMPIRICAL FORMULA)

C G(mil gas) (103) = $KN_2 \cdot A/T_{mil}$

C G(mil seam) (-104) = $Leam \cdot Wseam \cdot Fseam \cdot STEF$

C G(mil pin) (105) = $Kpin \cdot Npin(ratio) \cdot A \cdot Apin/T_{mil}$ C G(mil pin) (105) = $Kpin \cdot Npin(ratio) \cdot A \cdot Apin/T_{mil}$

PROBLEM DESCRIPTION

• SINDA '85/Fluint Model FMLI.INP (cont)

C G(sofi) (106) = $K_{sofi} \cdot A / (T_{sofi})^{1/2}$
 C G(sofi) (107) = $K_{sofi} \cdot A / (T_{sofi})^{1/2}$
 C G(wall-shield) (108) = $A \cdot F_v \cdot F_e \cdot STEF$
 C G(GN2-SHIELD) (109) = $h \cdot A$

C

-100,	10,	20,	1.0	\$ SHIELD TO MLI
101,	20,	30,	1.0	\$ MLI CONDUCTION
102,	20,	30,	1.0	\$ MLI RADIATION
SIV 103,	20,	30,	A3, K501	\$ MLI GAS
-104,	20,	30,	1.0	\$ MLI SEAM
SPV 105,	20,	30,	A1, K502	\$ MLI PIN
SPV 106,	30,	40,	A2, K503	\$ SOFI CONDUCTION
SPV 107,	40,	50,	A2, K504	\$ SOFI CONDUCTION
-108,	10,	8,	1.0	\$ WALL TO SHIELD
109,	10,	9,	1.0	\$ GN2 CONV SHIELD-WALL

HEADER ARRAY DATA, THIN

1=3.23365E-2,3.35183E-4,-4.6414E-7,3.23797E-10 \$ KPIN
 2=0.00259,0.0000231 \$ K SOFI BX260
 3= 115.0, 0.094 \$ K (GN2) 139.1, 0.0787
 139.2, 0.00439
 460.0, 0.0131
 800.0, 0.0204
 1000.0, 0.0243

HEADER CARRAY DATA, THIN

999=PARAMETER

HEADER FLOW DATA,PURGE,FID=8728

LU PLEN,10,PL = 14.7, TL = 530.
 LU JUNC,1, PL = 14.7, TL = 530.
 LU PLEN,20,PL = 14.7, TL = 530.
 PA CONN,1,10,1 \$ SEE OPERATIONS BLOCK
 DEV = MFRSET
 SMFR = 1.0
 PA CONN,2,1,20 \$ SEE OPERATIONS BLOCK
 DEV = STUBE
 TLEN = 1.0 DH = 1.0
 AF = 1.0
 T HTN,1,1,THIN.10,2,0.5
 T HTN,2,1,THIN.20,2,0.5

PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP (cont)

HEADER OPERATION DATA

```

BUILD TEST,THIN
BUILDF TEST,PURGE
777  CONTINUE ←
      XMLTN = XLAYTN/DMLITN/12.0
C
C WRITE INPUT PARAMETERS TO SCREEN
  WRITE(2,1100)CASE
C CRYO PHYSICAL PROPERTIES:
  WRITE(2,1101)TCRYO,QEVAP
C BOUNDARY CONDITIONS:
  WRITE(2,1102)TWALL,TSHIELD,GN2PT
C MLI PROPERTIES:
  WRITE(2,1103)AMLITN,DMLITN,XLAYTN,
    1 XMLTN,EMLIH,EMLIO,PDIA,
    1 DPIN,FMSTN,SEAMLTN,SEAMWTN
C SOFI PROPERTIES:
  WRITE(2,1104)ASOFTN,SOFITN
C WALL PROPERTIES:
  WRITE(2,1105)EWALLC SHIELD PROPERTIES:
  WRITE(2,1106)ASHTN,ESHTD,FSW,CONVSW
C GN2 PURGE FLUINT NETWORK
  WRITE(2,1108)GN2MLIT,GN2PFR
C
9995  WRITE(2,9994) ←
9994  FORMAT(/
    1 ' ENTER PARAMETER NAME TO BE CHANGED: (EX: TCRYO)/
    1 ' TO RUN WITH CHANGES ENTER: RUN/
    1 ' TO QUIT ENTER: QUIT OR EXIT')
  READ(1,'(A)',END=9995)THIN.UCA999
  IF(THIN.UCA999(1:1).EQ.' ')GO TO 9995
  IF(THIN.UCA999(1:3).EQ.'RUN')GO TO 778
  IF(THIN.UCA999(1:4).EQ.'QUIT')GO TO 779
  IF(THIN.UCA999(1:4).EQ.'EXIT')GO TO 779
9996  WRITE(2,9997)THIN.UCA999
9997  FORMAT(/
    1 ' ENTER VALUE FOR ',A8)
  READ(1,'*',END=9998)VALUE
C
  IF(THIN.UCA999(1:5).EQ.'QEVAP' )QEVAP=VALUE
  IF(THIN.UCA999(1:5).EQ.'TCRYO' )TCRYO=VALUE
C BOUNDARY CONDITIONS:
  IF(THIN.UCA999(1:5).EQ.'TWALL' )TWALL=VALUE
  IF(THIN.UCA999(1:7).EQ.'TSHIELD')TSHIELD=VALUE
  IF(THIN.UCA999(1:5).EQ.'GN2PT')GN2PT=VALUE

```

PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP (cont)

```

C MLI PROPERTIES:
  IF(THIN.UCA999(1:6).EQ.'AMLITN')AMLTN=VALUE
  IF(THIN.UCA999(1:6).EQ.'DMLITN')DMLTN=VALUE
  IF(THIN.UCA999(1:6).EQ.'XLAYTN')XLAYTN=VALUE
  IF(THIN.UCA999(1:5).EQ.'EMLIH')EMLIH=VALUE
  IF(THIN.UCA999(1:5).EQ.'EMLIO')EMLIO=VALUE
  IF(THIN.UCA999(1:4).EQ.'PDIA')PDIA=VALUE
  IF(THIN.UCA999(1:4).EQ.'DPIN')DPIN=VALUE
  IF(THIN.UCA999(1:5).EQ.'FMSTN')FMSTN=VALUE
  IF(THIN.UCA999(1:7).EQ.'SEAMLTN')SEAMLTN=VALUE
  IF(THIN.UCA999(1:7).EQ.'SEAMWTN')SEAMWTN=VALUE
C SOFI PROPERTIES:
  IF(THIN.UCA999(1:7).EQ.'ASOFTN')ASOFTN=VALUE
  IF(THIN.UCA999(1:6).EQ.'SOFTN')SOFTN=VALUE
C WALL PROPERTIES:
  IF(THIN.UCA999(1:5).EQ.'EWALL')EWALL=VALUE
C SHIELD PROPERTIES:
  IF(THIN.UCA999(1:5).EQ.'ASHTN')ASHTN=VALUE
  IF(THIN.UCA999(1:7).EQ.'ESHIELD')ESHIELD=VALUE
  IF(THIN.UCA999(1:3).EQ.'FSW')FSW=VALUE
  IF(THIN.UCA999(1:6).EQ.'CONVSW')CONVSW=VALUE
<C GN2 PURGE FLUINT NETWORK
  IF(THIN.UCA999(1:7).EQ.'GN2MLIT')GN2MLIT=VALUE
  IF(THIN.UCA999(1:6).EQ.'GN2PFR')GN2PFR=VALUE

C
  GO TO 777
C
778  CONTINUE
C
C WRITE INPUT PARAMETERS TO USER1 FILE
  WRITE(NUSER1,100)CASE
C CRYO PHYSICAL PROPERTIES:
  WRITE(NUSER1,101)TCRYO,QEVAP
C BOUNDARY CONDITIONS:
  WRITE(NUSER1,102)TWALL,TSIELD,GN2PT
C MLI PROPERTIES:
  WRITE(NUSER1,103)AMLTN,DMLTN,XLAYTN,
  1 XMLTN,EMLIH,EMLIO,PDIA,
  1 DPIN,FMSTN,SEAMLTN,SEAMWTN
C SOFI PROPERTIES:
  WRITE(NUSER1,104)ASOFTN,SOFTN
C WALL PROPERTIES:
  WRITE(NUSER1,105)EWALL
C SHIELD PROPERTIES:
  WRITE(NUSER1,106)ASHTN,ESHIELD,FSW,CONVSW
C GN2 PURGE FLUINT NETWORK
  WRITE(NUSER1,108)GN2MLIT,GN2PFR
  
```

```

graph TD
    A[GO TO 777] --> B[CONTINUE]
    B --> A
  
```

PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP (cont)

```

CALL CHGLMP('PURGE',10,'TL',GN2MLT,'PL')
PURGE.SMFR1 = GN2PFR*60.0
C THIN MLI WETTED HEAT TRANSFER AREA = 4.0*TLEN*AF/DH
PURGE.DH2 = 3.5-3.0
PURGE.AF2 = PI/4.*(3.5**2-3.0**2)
PURGE.TLEN2 = (2.*AMLITN)*PURGE.DH2/4/PURGE.AF2

THIN.T8 = TWALL
THIN.T9 = GN2PT
THIN.T50 = TCRYO
CALL HNQCAL('THICK')
CALL STDSTL
THIN.XK888 = -THIN.Q50
THIN.XK999 = THIN.XK888/QEVAP

C WRITE TEMPERATURE OUTPUT TO USER1 FILE
WRITE(NUSE1,201)
1 THIN.T50,THIN.T30,THIN.T20,THIN.T10,THIN.T8
WRITE(NUSE1,203) 1 PURGE.FR2/60.,PURGE.TL1,PURGE.TL1,PURGE.PL1,
1 HTCTN,VPURTN,RENTN

WRITE(NUSE1,301)
1 THIN.XK888,THIN.XK999

ICASE = ICASE+1
GO TO 777
779 CONTINUE

100 FORMAT(
1 '——GROUND HOLD TEST PREDICTIONS——'//
1 '—INPUT PARAMETERS FOR CASE NO.:J4, -/'
1 '———')
101 FORMAT(
1 'CRYO PHYSICAL PROPERTIES :'/
1 'TCRYO =',E10.4,' CRYO TANK TEMP (DEG R)'/
1 'QEVAP =',E10.4,' HEAT OF VAPORIZATION OF CRYO (BTU/LBM)')
102 FORMAT(/
1 'BOUNDARY CONDITIONS :'/
1 'TWALL =',E10.4,' CHAMBER WALL AND PLATFORM TEMP (DEG R)'/
1 'TSHIELD =',E10.4,' ALUMINUM SHIELD TEMP (DEG R)'/
1 'GN2PT =',E10.4,' CHAMBER GNE PURGE TEMP (DEG R)')
103 FORMAT(/ 1 'MLI PROPERTIES :'/
1 'AMLITN =',E10.4,' THIN MLI SURFACE AREA'/
1 'DMLITN =',E10.4,' THIN MLI DENSITY (LAYERS/IN)'/
1 'XLAYTN =',E10.4,' NO. OF MLI LAYERS ON THIN + 2'/
1 'XMLITN =',E10.4,' THIN MLI THICKNESS (FT) (OUTPUT ONLY)'/
1 'EMLIH =',E10.4,' MLI HEMISPHERICAL EMISIVITY'/
1 'EMLIO =',E10.4,' MLI OUTER LAYER EMISIVITY'/
1 'PDIA =',E10.4,' LEXAN PIN DIA (FT)'/

```

PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP (cont)

```

1 'DPIN = 'E10.4,' PIN DENSITY (NO./SQFT)'/
1 'FMSTN = 'E10.4,' THIN MLI-SHIELD VIEW FACTOR'/
1 'SEAMLTN = 'E10.4,' SEAM LENGTH (FT)'/
1 'SEAMWTN = 'E10.4,' SEAM WIDTH (FT)'/
104 FORMAT(/
1 'SOFI PROPERTIES :'/
1 'ASOFITN = 'E10.4,' THIN SOFI AREA (FT2)'/
1 'SOFITN = 'E10.4,' THIN SOFI THICKNESS (FT)'/
105 FORMAT(/
1 'VACUUM CHAMBER WALL PROPERTIES :'/
1 'EWALL = 'E10.4,' EMISSIVITY OF VAC WALL CHAMBER'/
106 FORMAT(/
1 'SHIELD PROPERTIES :'/
1 'ASHTN = 'E10.4,' SHIELD AREA (FT2)'/
1 'ESHIELD = 'E10.4,' EMISSIVITY OF AL SHIELD'/
1 'FSW = 'E10.4,' SHIELD-WALL VIEW FACTOR'/
1 'CONVSW = 'E10.4,' GN2 CONV SHIELD-WALL (BTU/HR-FT2-F)/
C
C GN2 PURGE FLUINT NETWORK
108 FORMAT(
1 'GN2 PURGE FLUINT NETWORK :'/ 1 'GN2MLIT = 'E10.4,' MLI GN2 PURGE GAS TEMP
(DEG R)'/
1 'GN2PFR = 'E10.4,' MLI GN2 PURGE FLOWRATE (LBS/MIN)'/
C
201 FORMAT(/
1 'FOAM/MLI TEMPERATURES (DEG R) BY SUB-MODELS :'/
1 '_____'/'
1 'THIN MLI :',
1 'CRYO SOFI MLI AL SHIELD',
1 'WALL'/,5(2X,F8.2))
C
C SUBMODEL PURGE OUTPUT
C
203 FORMAT(
1 'THIN PURGE FLOW NETWORK INFO'/
1 '_____'/'
1 'FR2(LB/MIN) TL1 (F) TL2 (F) PL2 (PSI)'
1 'H(B/HR-FT2-F) V(FT/SEC) REN NO. '/
1 4(1X,E10.5),3(2X,E10.5))
C
301 FORMAT(/
1 'HEAT LEAK BOIL-OFF RATE/ 1 ' (BTU/HR) (LBS/HR) '/
1 '_____'/'
1 'THIN 'E12.4,' 'E10.4,/)
C
1100 FORMAT(/// GROUND HOLD TEST PREDICTIONS INPUT PARAMETERS'
1 'FOR CASE NO. 'J4)
C

```


PROBLEM DESCRIPTION

• SINDA'85/Fluint Model FMLI.INP (cont)

```

1101 FORMAT(
1 'TCRYO =',E10.4,' QEVAP =',E10.4)
1102 FORMAT(
1 'TWALL =',E10.4,' TSHIELD =',E10.4/
1 'GN2PT =',E10.4)
1103 FORMAT(
1 'AMLITN =',E10.4,' DMLITN =',E10.4/
1 'XLAYTN =',E10.4,' XMLITN =',E10.4/
1 'EMLIH =',E10.4,' EMLJO =',E10.4/
1 'PDIA =',E10.4,' DPIN =',E10.4/
1 'FMSTN =',E10.4,' SEAMLITN =',E10.4/
1 'SEAMWTN =',E10.4)
1104 FORMAT(
1 'ASOFTN =',E10.4,' SOFITN =',E10.4)
1105 FORMAT(
1 'EWALL =',E10.4)
1106 FORMAT(
1 'ASHTN =',E10.4,' ESHIELD =',E10.4/
1 'FSW =',E10.4,' CONVSU =',E10.4)
C GN2 PURGE FLUINT NETWORK
1108 FORMAT( 1 'GN2MLIT =',E10.4,' GN2PFR =',E10.4)
C
C HEADER VARIABLES 1, THIN

C CALCULATING VALUES FOR BASIC MLI HEAT LEAK COMPONENTS
C Q(mli cond) = [A*8.95E-8*NLC^2.56/(2*(N-1))]*[Tm^2 - Ts^2]
C Q(mli rad) = [A*5.39E-10*etoth/(N-1)]*[Tm^4.67 - Ts^4.67]
C
C NOTE: THE ABOVE EQUATIONS UTILIZE SI UNITS B/C EQUATIONS ARE GIVEN
C AS SUCH.
C
C CONVERSION FACTORS WERE USED FOR CONTINUITY OF INPUTS.
C AREA: 1 FT**2 = .092903 M**2
C LENGTH: 1 IN = 2.54 CM
C HEAT: 1 BTU/HR = .29307 WATTS
C TEMPERATURE: DEG R = T(DEG R) = 1.8 T(DEG K)
C G100 = AMLITN*FMSTN*(1/(1/EMLJO+1/ESHIELD-1.))*STEF
XK201 = 8.95E-8*((DMLITN/2.54)**2.56)/(2*(XLAYTN-1.))
XK102 = (AMLITN*.092903*XK201)*((T20/1.8)**2-(T30/1.8)**2)/.29307
G101 = XK102/(T20-T30)
XK301 = 5.39E-10*EMLIH/(XLAYTN-1.)
XK101 = (AMLITN*.092903*XK301)*((T20/1.8)**4.67-(T30/1.8)**4.67)/.29307
G102 = XK101/(T20-T30)
XK501 = AMLITN/XMLITN

```

PROBLEM DESCRIPTION

• SINDA '85/Fluint Model FMLI.INP (cont)

```
G104 = SEAMLTN*SEAMWTN*(SQRT(1.+XMLITN**2/SEAMWTN**2)-XMLITN/SEAMWTN)*
XK502 = DPIN*AMLITN*PI*PDIA**2/XMLITN
XK503 = ASOFTN/(SOFTN/2.)
XK504 = XK503
G108 = ASHTN*FSW*(1./(1./ESHIELD+1./EWALL-1.))*STEF
G109 = CONVSW*ASHTN
```

HEADER OUTPUT CALLS, THIN

```
IF(LOOPCT.GT.1) THEN
CALL TPRINT ('THIN')
CALL HNQPNT ('THIN')
END IF
```

HEADER OUTPUT CALLS, PURGE

```
IF(LOOPCT.GT.1) THEN
CALL LMPTAB ('PURGE')
CALL TIETAB ('PURGE')
CALL PHTTAB ('PURGE')
END IF
```

<

HEADER FLOGIC 1, PURGE

C OBTAIN PURGE GAS V AND R_o BETWEEN SHIELD AND THIN MLI (FT/SEC)

C V = MDOT * SPEC VOL / FLOW AREA

VPURTN = PURGE.SMFR1*VSV(PL1,TL1,PURGE.FI)/PURGE.AF2/3600.

HTCTN = DITTUS(PURGE.FR2,PURGE.DH2,PURGE.AF2,THIN.T20,PURGE.PL1,

1 PURGE.TL1,PURGE.XL1,PURGE.FI)

RENTN = VPURTN*3600.*PURGE.DH2/VSV(PL1,TL1,PURGE.FI)

1 VVISCV(PL1,TL1,PURGE.FI)

C

END OF DATA

EXAMPLE PROBLEM

• Interactive Run: Compile and Link Input File FMLLINP

```

EPVAX>dir
Directory DISK(USER4:[SIMMONDS.WORK.TFAWS])

FMLLINP;3

Total of 1 file, 31 blocks.
EPVAX>etndm85save fmllinp
INPUT DATA FILE: FMLLINP
SINDA '85 PREPROCESSOR RUN OF PROBLEM: FMLLINP
STARTING: 11-AUG-1982 07:37:20.31
FORTRAN STOP
The Pre-Processor ran for 6 cpu seconds
PREPROCESSOR ENDS : 11-AUG-1982 07:37:55.84
BEGINNING COMPILE AND LINK
STARTING: 11-AUG-1982 07:37:55.85
The compiler ran for 4 cpu seconds
DOING THE LINK
The link ran for 7 cpu seconds
COMPILE AND LINK ENDS : 11-AUG-1982 07:38:23.33
SINDA '85 PROCESSOR RUN OF PROBLEM: FMLLINP
STARTING: 11-AUG-1982 07:38:23.73
EPVAX>

```

EXAMPLE PROBLEM

• Interactive Run: Output Files (Created in Scratch Directory ZZZZZZ.DIR)

```

EPVAX>dir
Directory DISK(USER4:[SIMMONDS.WORK.TFAWS.ZZZZZZ])

ARYDAT.DAT;1  ARYTRE.DAT;1  CARTRE.DAT;1  CNTDAT.DAT;1
CNTTRE.DAT;1  CONNAM.DAT;1  CONTRE.DAT;1  CRYDAT.DAT;1
FLOCON.DAT;1  FLODEV.DAT;1  FLOJUN.DAT;1  FLOMOO.DAT;1
FLOPLN.DAT;1  FLOPRP.DAT;1  FLOTIE.DAT;1  FLOTNK.DAT;1
FLOTUS.DAT;1  FMILLEKE;1  LMPTRE.DAT;1  NODDAT.DAT;1
NODTRE.DAT;1  NOUSER.DAT;1  NUMTRE.DAT;1  NVCDAT.DAT;1
NVQDAT.DAT;1  OPTION.DAT;1  OPTIONS.DAT;1  PCSDAT.DAT;1
SORDAT.DAT;1  TIE TRE.DAT;1  TRYDAT.DAT;1  USBDAT.DAT;1

Total of 32 files.
EPVAX>

```

EXAMPLE PROBLEM

• Interactive Run: Running Interactive (File FMLI.EXE)

EPVAX>run fml

GROUND HOLD TEST PREDICTIONS INPUT PARAMETERS FOR CASE NO. 1

TCRYO = 0.3700E+02 QEVAP = 0.1918E+03
 TWALL = 0.5300E+03 TSHIELD = 0.5200E+03
 GN2PT = 0.5300E+03 DMLJTN = 0.5000E+02
 AMLJTN = 0.4581E+02 XMLJTN = 0.2833E+01
 XLAYTN = 0.1700E+02 EMLJN = 0.5000E-01
 EMLJO = 0.1000E+00
 PDIA = 0.2804E-02 DPIN = 0.1000E+01
 FMSTN = 0.1000E+01 SEAMLTN = 0.2806E+02
 SEAMWTH = 0.1042E-01
 ASOFTN = 0.4399E+02 SOFTN = 0.3750E-01
 EWALL = 0.8000E+00
 ASHTN = 0.5203E+02 ESHIELD = 0.8000E+00
 FBW = 0.1000E+01 CONVSF = 0.1000E+01
 GN2MLJT = 0.5300E+03 GN2PFR = 0.1000E+02

ENTER PARAMETER NAME TO BE CHANGED: (EX: TCRYO)
 TO RUN WITH CHANGES ENTER: RUN
 TO QUIT ENTER: QUIT OR EXIT

EXAMPLE PROBLEM

• Interactive Run: Entering Inputs (Modify TCRYO User Data to 140)

TCRYO

ENTER VALUE FOR TCRYO
 140

GROUND HOLD TEST PREDICTIONS INPUT PARAMETERS FOR CASE NO. 1

TCRYO = 0.1400E+03 QEVAP = 0.1918E+03
 TWALL = 0.5300E+03 TSHIELD = 0.5200E+03
 GN2PT = 0.5300E+03 DMLJTN = 0.5000E+02
 AMLJTN = 0.4581E+02 XMLJTN = 0.2833E+01
 XLAYTN = 0.1700E+02 EMLJN = 0.5000E-01
 EMLJO = 0.1000E+00
 PDIA = 0.2804E-02 DPIN = 0.1000E+01
 FMSTN = 0.1000E+01 SEAMLTN = 0.2806E+02
 SEAMWTH = 0.1042E-01
 ASOFTN = 0.4399E+02 SOFTN = 0.3750E-01
 EWALL = 0.8000E+00
 ASHTN = 0.5203E+02 ESHIELD = 0.8000E+00
 FBW = 0.1000E+01 CONVSF = 0.1000E+01
 GN2MLJT = 0.5300E+03 GN2PFR = 0.1000E+02

ENTER PARAMETER NAME TO BE CHANGED: (EX: TCRYO)
 TO RUN WITH CHANGES ENTER: RUN
 TO QUIT ENTER: QUIT OR EXIT

EXAMPLE PROBLEM

- Interactive Run:
Entering Inputs (Modify CONVW User Constant to .5)

```
CONVSW

ENTER VALUE FOR CONVSW
.5

GROUND HOLD TEST PREDICTIONS INPUT PARAMETERS FOR CASE NO. 1
TCRYO = 0.1400E+03 QEVAP = 0.1010E+03
TWALL = 0.5300E+03 TSHIELD = 0.5200E+03
GN2PT = 0.5300E+03
AMLITN = 0.4581E+02 DMLITN = 0.5000E+02
XLAYTN = 0.1700E+02 XMLITN = 0.2833E-01
EMLIN = 0.5000E-01 EMLJO = 0.1000E+00
POLA = 0.2804E-02 DFIN = 0.1000E+01
FMSTN = 0.1000E+01 SEAMLTN = 0.3800E+02
SEAMWTN = 0.1042E-01
ASOFTN = 0.4300E+02 SOFTN = 0.3750E-01
EWALL = 0.8000E+00
ASHTN = 0.6203E+02 ESHIELD = 0.9000E+00
FSW = 0.1000E+01 CONVSW = 0.5000E+00
GN2MLJT = 0.5300E+03 GN2PFR = 0.1000E+02

ENTER PARAMETER NAME TO BE CHANGED: (EX: TCRYO)
TO RUN WITH CHANGES ENTER: RUN
TO QUIT ENTER: QUIT OR EXIT
```

EXAMPLE PROBLEM

- Interactive Run:
Run Case1

```
RUN

GROUND HOLD TEST PREDICTIONS INPUT PARAMETERS FOR CASE NO. 2
TCRYO = 0.1400E+03 QEVAP = 0.1010E+03
TWALL = 0.5300E+03 TSHIELD = 0.5200E+03
GN2PT = 0.5300E+03
AMLITN = 0.4581E+02 DMLITN = 0.5000E+02
XLAYTN = 0.1700E+02 XMLITN = 0.2833E-01
EMLIN = 0.5000E-01 EMLJO = 0.1000E+00
POLA = 0.2804E-02 DFIN = 0.1000E+01
FMSTN = 0.1000E+01 SEAMLTN = 0.3800E+02
SEAMWTN = 0.1042E-01
ASOFTN = 0.4300E+02 SOFTN = 0.3750E-01
EWALL = 0.8000E+00
ASHTN = 0.6203E+02 ESHIELD = 0.9000E+00
FSW = 0.1000E+01 CONVSW = 0.5000E+00
GN2MLJT = 0.5300E+03 GN2PFR = 0.1000E+02

ENTER PARAMETER NAME TO BE CHANGED: (EX: TCRYO)
TO RUN WITH CHANGES ENTER: RUN
TO QUIT ENTER: QUIT OR EXIT
```

EXAMPLE PROBLEM

- **Interactive Run:
Exit and Show Result Files in TFAWS.DIR**

```
EXIT
FORTRAN STOP
EPVAX>dir [-]
Directory DISK$USER4:[SIMMONDS.WORK.TFAWS]

FMLLINP;3    FMLLUS;1    FMLLOPP;1    FMLLOUT;1
FMLLUS1;1    TFAWS.DIR;1

Total of 6 files.
EPVAX>
```

SUMMARY

- **The Interactive Process Saves Time.**
- **Permits Modifications to Thermal/Fluids Model Parameters During Run Time.**
- **Permits User to Examine Results and Make Decisions During Parametric Studies.**
- **Executable Models Can be Run by Non-SINDA '85/Fluint Users.**
- **Open the Doors for Unlimited Creativity and Interaction with the SINDA '85/Fluint Models.**

SUMMARY

• Output File FMLI.US1

—GROUND HOLD TEST PREDICTIONS—

—INPUT PARAMETERS FOR CASE NO. 1 —

CRYO PHYSICAL PROPERTIES :

TCRYO = 0.1400E+03 CRYO TANK TEMP (DEG R)

QEVP = 0.1919E+03 HEAT OF VAPORIZATION OF CRYO (BTU/LBM)

BOUNDARY CONDITIONS :

TWALL = 0.5300E+03 CHAMBER WALL AND PLATFORM TEMP (DEG R)

TSHIELD = 0.5200E+03 ALUMINUM SHIELD TEMP (DEG R)

GN2PT = 0.5300E+03 CHAMBER GNE PURGE TEMP (DEG R)

MLI PROPERTIES :

AMLITN = 0.4581E+02 THIN MLI SURFACE AREA

DMLTN = 0.5000E+02 THIN MLI DENSITY (LAYERS/IN)

XLAYTN = 0.1700E+02 NO. OF MLI LAYERS ON THIN + 2

XMLITN = 0.2833E-01 THIN MLI THICKNESS (FT) (OUTPUT ONLY)

EMLIH = 0.5000E-01 MLI HEMISPHERICAL EMISSIVITY

EMLJO = 0.1000E+00 MLI OUTER LAYER EMISSIVITY

PDIA = 0.2604E-02 LEXAN PIN DIA (FT)

DPIN = 0.1000E+01 PIN DENSITY (NO./SQFT)

FMSTN = 0.1000E+01 THIN MLI-SHIELD VIEW FACTOR

SEAMLTN = 0.3606E+02 SEAM LENGTH (FT)

SEAMWTN = 0.1042E-01 SEAM WIDTH (FT)

SOFI PROPERTIES :

ASOFITN = 0.4399E+02 THIN SOFI AREA (FT²)

SOFITN = 0.3750E-01 THIN SOFI THICKNESS (FT)

VACUUM CHAMBER WALL PROPERTIES :

EWALL = 0.8000E+00 EMISSIVITY OF VAC WALL CHAMBER

SHIELD PROPERTIES :

ASHTN = 0.6203E+02 SHIELD AREA (FT²)

ESHIELD = 0.9000E+00 EMISSIVITY OF AL SHIELD

FSW = 0.1000E+01 SHIELD-WALL VIEW FACTOR

CONVSW = 0.5000E+00 GN2 CONV SHIELD-WALL (BTU/HR-FT²-F)

GN2 PURGE FLUENT NETWORK :

GN2MLT = 0.5300E+03 MLI GN2 PURGE GAS TEMP (DEG R)

GN2PFR = 0.1000E+02 MLI GN2 PURGE FLOWRATE (LBS/MIN)

FOAM/MLI TEMPERATURES (DEG R) BY SUB-MODELS :

THIN MLI :

CRYO SOFI MLI AL SHIELD WALL

140.00 312.56 406.56 524.75 530.00

THIN PURGE FLOW NETWORK INFO

FR2(LB/MIN) TL1 (F) TL2 (F) PL2 (PSI) H(B/HR-FT²-F) V(FT/SEC) REN NO.

.10000E+02 .52208E+03 .52208E+03 .14700E+02 .22781E+00 .88730E+00 .27915E+04

HEAT LEAK BOIL-OFF RATE

(BTU/HR) (LBS/HR)

THIN 0.1582E+04 0.8245E+01

SUMMARY

• VAX FILES (Command Procedures) to Run SINDA'85

EPVAX>d

Directory DISK\$USER4:[SIMMONDS.THERMAL.SINDA85]

ASTA.COM;20	7 16-OCT-1991 13:25:41.10 (RWED,RWED,RE,RE)
ASTAPP.EXE;1	902 19-APR-1990 14:55:33.00 (RWED,RWED,RE,RE)
ASTASAVE.COM;3	7 22-APR-1991 15:37:31.10 (RWED,RWED,RE,RE)
BANNER.TXT;7	3 21-JUL-1989 12:54:24.00 (RWED,RWED,RE,RE)
BANNER2.TXT;3	7 21-JUL-1989 12:54:35.00 (RWED,RWED,RE,RE)
DATA_ONLY.DIR;1	1 26-SEP-1990 07:51:58.12 (RWED,RWED,RE,RE)
DELWORK.COM;24	2 29-JUN-1989 16:34:40.00 (RWED,RWED,RE,RE)
EXPLOT.DIR;1	1 25-SEP-1990 16:18:12.00 (RWED,RWED,RE,RE)
FINCLUDE.DIR;1	3 26-SEP-1990 11:20:28.10 (RWED,RWED,RE,RE)
FLUINTP.OLB;2	2204 19-APR-1990 14:20:42.00 (RWED,RWED,RE,RE)
FLUINTPP.OLB;1	1480 19-APR-1990 14:19:31.00 (RWED,RWED,RE,RE)
FS_ROUTINES.OLB;1	116 4-OCT-1991 14:19:03.27 (RWED,RWED,RE,RE)
INCLUDE.DIR;1	4 26-SEP-1990 11:20:22.93 (RWED,RWED,RE,RE)
LINKPPF.COM;3	1 24-JUL-1988 11:16:38.00 (RWED,RWED,RE,RE)
LINKRAP.COM;2	1 6-SEP-1989 08:15:49.00 (RWED,RWED,RE,RE)
MKNAME.COM;1	1 15-AUG-1984 10:30:06.00 (RWED,RWED,RE,RE)
MKWORK.COM;17	2 29-JUN-1989 08:33:50.00 (RWED,RWED,RE,RE)
NEW_FS_ROUTINES.OLB;1	
	48 19-FEB-1991 13:23:30.60 (RWED,RWED,RE,RE)
OLD_FS_ROUTINES.OLB;1	
	116 20-JUL-1989 16:24:38.00 (RWED,RWED,RE,RE)
OLD_TSAVE_ASCII.OBJ;1	
	3 10-OCT-1990 11:56:28.20 (RWED,RWED,RE,RE)
SAMPLES.DIR;1	1 10-AUG-1990 09:55:56.44 (RWED,RWED,RE,RE)
SETHOME.COM;7	1 29-JUN-1989 08:28:58.00 (RWED,RWED,RE,RE)
SINDA85.COM;24	12 20-APR-1990 10:21:41.00 (RWED,RWED,RE,RE)
SINDA85.USAGE;6	20 9-JUL-1992 13:28:21.82 (RWED,RWED,RWE,RWE)
SINDA85SAVE.USAGE;4	
	1 9-JUL-1992 13:28:44.40 (RWED,RWED,RWE,RWE)
TSAVE_ASCII.OBJ;4	3 19-FEB-1991 13:30:13.31 (RWED,RWED,RE,RE)
UTILITY.OLB;1	113 19-APR-1990 14:18:43.00 (RWED,RWED,RE,RE)

Total of 27 files, 5060 blocks.

EPVAX>

SUMMARY

• ASTASAVE.COM With Minimum Modifications Required

```

$ SET NOVERI
$ SET WORK/LIMIT=1024
$ On WARNING then goto EXIT
$ On CONTROL_Y then goto EXIT
$ ASSIGN $1$due4:[user.SIMMONDS.THERMAL.SINDA85] AST
$!
$!
$ node = f$getsys("NODENAME") - "SYS"
$ start_time = f$time()
$ start_cpu = f$getjpl("", "cputim")
$!
$! GET THE CURRENT LOCATION AND PLACE TO KEEP THE RESULTS
$!
$ IF P1 .NES. "" THEN GOTO ISINPUT
$ WRITE SYS$OUTPUT " ***** ERROR - NO INPUT "
$ GOTO EXIT
$ ISINPUT:
$ WRITE SYS$OUTPUT "INPUT DATA FILE: "P1"
$ @AST:SETHOME 'P1
$!
$ FNAME = F$PARSE(P1, "NAME") + F$PARSE(P1, "TYPE")$ ASSIGN
'SINDA85_KEEP_DIR"FNAME FOR005
$ FNAME = F$PARSE(P1, "NAME")
$ Assign 'SINDA85_KEEP_DIR"FNAME.OPP FOR005
$!
$! Assign the MITAS Processor TSAVE Plot file.
$ Assign 'SINDA85_KEEP_DIR"FNAME' KEEP$FILE
$!
$! CREATE WORKING DIRECTORY ON SCRATCH OR LOCALLY
$!
$ SET NOCONTROL=Y
$ @AST:MKWORK
$ SET CONTROL=Y
$!
$! RUN THE PRE PROCESSOR
$!
$ On WARNING then goto EXIT
$ On CONTROL_Y then goto EXIT
$ WRITE SYS$OUTPUT "SINDA '85 PREPROCESSOR RUN OF PROBLEM: "P1"
$ WRITE SYS$OUTPUT "STARTING: "F$TIME()"
$ T1 = F$GETJPL("", "CPUTIM")
$! RUN/NODEB AST:flulntPP$ RUN/NODEB AST:ASTAPP
$ T2 = (F$GETJPL("", "CPUTIM") - T1)/100
$ WRITE SYS$OUTPUT " The Pre-Processor ran for "T2' cpu seconds"
$ WRITE SYS$OUTPUT "PREPROCESSOR ENDS : "F$TIME()"
$ DEASSIGN FOR005

```

SUMMARY

• ASTASAVE.COM With Minimum Modifications Required (cont)

```

$!
$!
$!   COMPILE AND LINK
$!
$ On WARNING then goto EXIT
$ On CONTROL_Y then goto EXIT
$ WRITE SYS$OUTPUT "BEGINNING COMPILE AND LINK"
$ WRITE SYS$OUTPUT "STARTING: "F$TIME()"
$ T1 = F$GETJPI("", "CPUTIM")
$ FOR/LIS='SINDA85_KEEP_DIR"FNAME.LIS/CROSS ASTAP.DAT
$ T2 = (F$GETJPI("", "CPUTIM") - T1)/100
$ WRITE SYS$OUTPUT " The compiler ran for "T2' cpu seconds"
$ Write SYS$OUTPUT " DOING THE LINK "
$ T1 = F$GETJPI("", "CPUTIM")
$ LINK/EXEC='FNAME.EXE ASTAP,AST:fluintp/L,UTILITY/L,FS_ROUTINES/L
$ T2 = (F$GETJPI("", "CPUTIM") - T1)/100
$ WRITE SYS$OUTPUT " The link ran for "T2' cpu seconds"
$ WRITE SYS$OUTPUT "COMPILE AND LINK ENDS : "F$TIME()"
$ DEL ASTAP.*;
$!
$!   RUN THE PROCESSOR
$!
$ On WARNING then goto EXIT
$ On CONTROL_Y then goto EXIT
$ ASSIGN 'SINDA85_KEEP_DIR"FNAME.TSV FOR021
$ ASSIGN 'SINDA85_KEEP_DIR"FNAME.RP FOR025
$ IF P2.EQS. "" THEN GOTO ENT1
$   WRITE SYS$OUTPUT "RSI DATA FILE: "P2'.RP"
$   @AST:MKNAME 'P2
$   PP2 = TNAME
$   ASSIGN 'PP2.RP FOR024
$ ENT1:
$ WRITE SYS$OUTPUT "SINDA '85 PROCESSOR RUN OF PROBLEM: "P1""
$ WRITE SYS$OUTPUT "STARTING: "F$TIME()"
$ T1 = F$GETJPI("", "CPUTIM")
$!
$ ASSIGN SYS$INPUT FOR001
$ ASSIGN SYS$OUTPUT FOR002
$!
$IRUN 'FNAME
$IT2 = (F$GETJPI("", "CPUTIM") - T1)/100
$!WRITE SYS$OUTPUT " The processor ran for "T2' cpu seconds"
$!WRITE SYS$OUTPUT "PROCESSOR ENDS : "F$TIME()"
$!
$!   DEL WORKING FILES AND DIRECTORIES
$!
$ EXIT:
$!

```

SUMMARY

• ASTASAVE.COM With Minimum Modifications Required (cont)

```
$ open/append usage ast:ainda85save.usage
$ write usage f$getip("","username"), " ","node", " ","start_time", -
    " ",f$time()
$ close usage
$!
$ On WARNING then continue
$ On CONTROL_Y then continue
$ On ERROR then continue
$ IF ""F$LOGICAL("FOR005")".NES."" THEN DEASSIGN FOR005
$ IF ""F$LOGICAL("FOR006")".NES."" THEN DEASSIGN FOR006
$!@AST:DELWORK
$ EXIT1:
```

TRASYS Form Factor Matrix Normalization

Glenn T. Tsuyuki*
 Jet Propulsion Laboratory
 California Institute of Technology
 Pasadena, California 91109

SUMMARY

A method has been developed for adjusting a TRASYS enclosure form factor matrix to unity. This approach is not limited to closed geometries, and in fact, it is primarily intended for use with open geometries. The purpose of this approach is to prevent optimistic form factors to space. In this method, nodal form factor sums are calculated within 0.05 of unity using TRASYS, although deviations as large as 0.10 may be acceptable, and then, a process is employed to distribute the difference amongst the nodes. A specific example has been analyzed with this method, and a comparison was performed with a standard approach for calculating radiation conductors. In this comparison, hot and cold case temperatures were determined. Exterior nodes exhibited temperature differences as large as 7°C and 3°C for the hot and cold cases, respectively when compared with the standard approach, while interior nodes demonstrated temperature differences from 0°C to 5°C. These results indicate that temperature predictions can be artificially biased if the form factor computation error is lumped into the individual form factors to space.

NOMENCLATURE

A_i	area of ith node
AU	astronomical units
BCS	block coordinate system
DDA	dual-drive actuator
FFCAL	form factor calculation segment within TRASYS
FFRATL	maximum internodal subelement distance to average internodal subelement distance ratio
F_{ij}	form factor from node i to node j
GLL	Galileo Project
GMM	geometric math model

* Technical Group Leader, Cassini Thermal Engineering Group

HGA	high gain antenna
LGA	low gain antenna
MLI	multilayer insulation
NELCT	number of subelements used in Nusselt unit sphere method
PWS	plasma wave science
S_i	nodal form factor sum for node i
S/C	spacecraft
TMM	thermal math model
TRASYS	Thermal Radiation Analyzer System
α_s	solar absorptivity
Δ_i	difference between nodal form factor sum for node i and unity
ϵ	hemispherical emissivity

Superscripts

cal	calculated directly through TRASYS
red	calculated through form factor reduction process
uni	calculated from process to adjust form factor matrix to unity

STATEMENT OF THE PROBLEM

TRASYS (Ref. 1) is a software system which is utilized for the determination of internodal form factors and environmental heating in primarily extraterrestrial thermal analyses. When GMMs are of moderate or large size, it becomes increasingly more difficult to verify their form factor calculations. Internodal shadowing and complex-shaped geometry are some reasons contributing to this obstacle. Thus, individual form factor verification is simply not practical for sizeable models. Of more pragmatic importance is the form factor from each node to space. TRASYS does not directly determine form factors to space in its standard operating mode. Instead, TRASYS implicitly uses the difference of the nodal form factor sum and unity. Therefore, any form factor computation error will be directly imbedded in the form factor to space. It should be noted that TRASYS possesses an option to enable direct calculation of the form factor to space. However, this option is computationally-intensive and has demonstrated computational errors (Refs. 2, 3, and 4). A more significant shortcoming with this approach is its inability to save form factor to space calculations on the restart file. Clearly, an approach that can address how the computational error is distributed over all nodes is required.

FORM FACTOR MATRIX NORMALIZATION

Standard Form Factor Calculation Mode

The FFCAL segment is responsible for form factor calculations within TRASYS. It is reliant upon a parameter known as FFRATL which represents the maximum internodal subelement distance to average internodal subelement distance ratio. The default value is 15.0, but it may

be respecified by the user. If the calculated FFRATL is less than the specified value, the double summation (numerical integration) technique is used for that particular F_{ij} . However, if the calculated FFRATL is greater than the specified value, the Nusselt unit sphere technique is employed. The Nusselt unit sphere technique is more accurate than the double summation method, but it is also more time-consuming as well. The default FFRATL value has been demonstrated to be an empirically optimal in terms of computation time and accuracy.

Creating Enclosures from Open Geometries

It has been indicated the individual form factors to space may be inherently erroneous if there is no provision to verify the calculation. A suggested approach is to construct an enclosure around the open geometry. This does not simply imply surrounding the geometry within a large sphere, but rather using appropriate-sized surfaces to complete closure. A simplistic example would be using a sufficiently-nodalized hemisphere to enclose a circular disk. The closing surfaces should be nodalized so that each enclosing nodal area is no more than one order of magnitude larger than the smallest node in the geometry, but ideally, it should be of the same magnitude. Such a constraint upon the enclosing area helps to ensure accurate form factor calculations to and from these nodes.

Optimizing Form Factor Calculations

In many instances, it is not tractable to determine the validity of every form factor calculation especially if the geometry does not constitute a complete enclosure. For enclosures, a more global but yet effective way of determining form factor calculation accuracy is the nodal form factor sum which must be unity. This idea may be extended to non-enclosures since it was previously explained how open geometries may be closed out. Usually, accuracy within ± 0.05 of unity is acceptable, but there may be cases where accuracy within ± 0.10 of unity is acceptable since temperature differences are expected to be small. Nodal form factor sums may not be acceptable even after the standard TRASYS form factor calculation procedure is implemented. Accuracy may be improved by recomputing individual form factors for those nodes whose form factor sums are unacceptable by forcing the Nusselt unit sphere technique and by using more nodal subelement resolution. In terms of application within TRASYS (see Fig. 1), the previous form factor calculation is restarted, recomputed nodes are identified through RECOMP option in the form factor data block, Nusselt unit sphere method is specified by setting FFRATL to -1.0, and higher nodal resolution is specified by setting NELCT equal to between 75 and 100 prior to the FFCAL call. With correctly-specified geometry, recomputation will usually bring form factor sums between 0.95 and 1.05.

Figure 1 - TRASYS run stream for form factor recomputation; italicized text indicates user input

```

HEADER OPTIONS DATA
  RSI $ READ RESTART TAPE FROM INITIAL FF RUN
  RSO $ WRITE RESTART OUTPUT TAPE

```

```

.
.
HEADER FORM FACTOR DATA
FIG  model configuration name
      node ID,RECOMP $ RECOMP FFs TO AND FROM THIS NODE
      node ID,RECOMP
.
.
.
HEADER OPERATIONS DATA
.
.
      NELCT = 100 $ SUBELEMENTAL BREAKDOWN SPECIFICATION
C*****
C*    USE UNIT SPHERE METHOD FOR FF RECOMP
C*****
      CALL FFDATA(value,value, -1.0,.....)
L    FFCAL
.
.
.
END OF DATA

```

If form factor recomputation does not produce acceptable nodal form factor sums, it would be advisable to reexamine the geometry for potential geometry problems such as gaps between nodes, inactive side of a node being viewed, or a node lying directly upon or intersecting another node.

Reducing Form Factor Sums Greater than Unity

Even after form factor recomputation, there may be a number of nodes whose form factor sums are unacceptably greater than unity. A simple algorithm has been devised to reduce the individual form factors on a weighted basis so that the nodal form factor sum is consequently reduced to or below unity. For any of the nodes in question, the difference from unity is determined as,

$$\Delta_i = \sum_j F_{ij}^{cal} - 1 \quad (1)$$

Or,

$$\Delta_i = S_i^{cal} - 1 \quad (2)$$

It is assumed that Δ_i represents the form factor computational error and furthermore, it is assumed that the error is proportional to the size of the nodal form factor. Hence, each nodal

$$F_{ij}^{red} = F_{ij}^{cal} - \frac{F_{ij}^{cal}}{\sum_j F_{ij}^{cal}} \Delta_i \quad (3)$$

form factor may be reduced based upon its fractional make-up of the form factor sum, and this weighing is demonstrated as the second term in Eq. 3. Eq. 3 may be rewritten as:

$$F_{ij}^{red} = \frac{F_{ij}^{cal}}{S_i^{cal}} \quad (4)$$

When the reduction process is complete, Eq. 4 indicates that the summation of the reduced nodal form factors should total unity. It should be noted that although the i th nodal form factor sum has been set to unity, the reduction process implicitly affects the j th nodal form factor sum due to form factor reciprocity. Consequently, there may be instances where the j th nodal form factor sum is perilously close to 0.95, and the reduction process will lead to an unacceptable form factor sum for the j th node. In these cases, this j th node should be excluded from the reduction process, and the weighing should be based on the remaining nodal form factors.

Adjusting Form Factor Matrix to Unity

Following the reduction process, the nodal form factor sums should not be greater than unity. It is possible to devise a process to increase form factor sums to unity at this point. However, the application of this process to every node would be difficult, because of the interdependency of the form factors through reciprocity. Instead, the main objective is to prevent the difference between the nodal form factor sum and unity from erroneously being added to the form factor to space. Therefore, the nodal form factor deviation from unity is assumed to be added to the form factor to itself (Eq. 5). Here, the implicit assumption is that there is virtually no

$$F_{ii}^{uni} = F_{ii}^{red} + (1 - \sum_j F_{ij}^{red}) \quad (5)$$

temperature differences between the nodes. Once Eq. 5 has been performed for all nodes, the form factor matrix should be entirely adjusted to unity. However, if this approach results in non-conservative modeling, an analogous form factor weighted process to increase form factor sums to unity may be applied to particular nodes of interest. Eq. 4 with a sign change would be applicable for this process.

It should be kept in mind that the enclosing nodes represent space. These enclosing surfaces may be removed from the GMM, and the form factors to space have been adjusted so that they are more rigorous in a global sense. In the form factor matrix normalization process, the computational error has been distributed throughout the GMM nodes. Therefore, the individual form factors to space do not have all the computational error imbedded in them.

Implementing the Normalized Form Factor Matrix

The GMM can be modified to remove the enclosing nodes. In order to facilitate removal, the

enclosing nodes should be specified in a separate BCS. Additionally, the form factor matrix must also be modified so that all form factors to or from the enclosing nodes are removed. The remaining form factors may be input through the form factor data block. This TRASYS run stream is depicted in Fig. 2. Note that an input restart file is not required since an entire form factor matrix is entered in the form factor data block. Also, note that the option to initially zero the entire form factor matrix is utilized since only non-zero form factors are input. This prevents TRASYS from calculating form factors that were known to be zero.

Figure 2 - Implementation of normalized form factors; italicized text indicates user input

```

HEADER OPTIONS DATA
.
.
  RSO $ WRITE AN OUTPUT RESTART FILE
.
.
HEADER SURFACE DATA
.
.
  geometry without enclosing surfaces
.
.
HEADER FORM FACTOR DATA
FIG  model configuration name
node array
ZERO $ INITIALLY SETS ENTIRE FA MATRIX TO ZERO
normalized form factors without enclosing surfaces
.
.
HEADER OPERATIONS DATA
.
.
L    FFCAL $ CALL TO FFDATA NOT NEED SINCE HEADER FORM FACTOR DATA USED
.
.
END OF DATA

```

Available Computer Codes for Normalization

A FORTRAN program known as PL-PULL (Ref. 5) has been developed by Rockwell International with the capability to normalize a form factor matrix as described above.

A SAMPLE APPLICATION

Form factor matrix normalization has been applied in the case of the GLL HGA GMM (Ref. 6). The hardware configuration is shown in Fig. 3, along with the GMM nodalization. The intent of this model is to be able to predict primarily exterior surface temperatures during its Venus flyby while in the stowed configuration, but internal components of interest such as the DDA and the S-band antenna feed have been also modeled. The TMM generally shares a one-to-one correspondence with the GMM with the exception of the ribs which are individually distinct and then collapsed into one bulk representation. This type of modeling is valid since the S/C is expected to be spinning about the axis of antenna symmetry when the HGA is stowed.

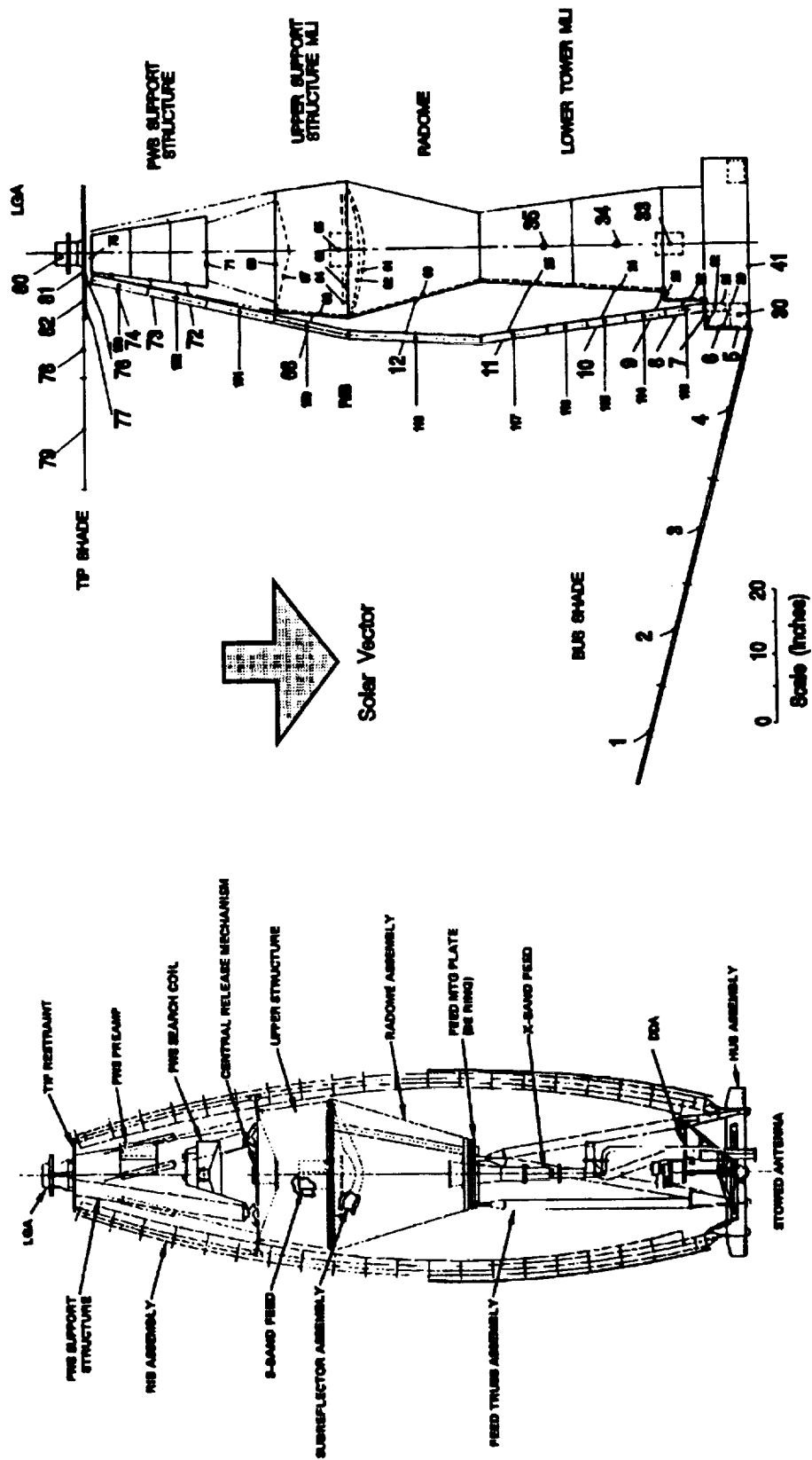


Figure 3 - GLL HGA hardware description (left) and model nodalization (right)

The external node descriptions are given in Table 1. The antenna is radiatively isolated from the rest of the S/C with an MLI blanket known as the bus shade. The lower tower is covered with MLI blankets as well as the stowed ribs and upper support structure. The radome and PWS support structure are covered with a single layer of black Kapton. The LGA is painted with white paint. The tip shade is carbon-filled Kapton and is used to provide protection from high solar irradiances. It should be noted that the DDA has a significant conductive tie with the S/C main body, and the main body is treated as a 25°C boundary temperature. For this sample problem, two extreme cases were investigated: 1) a hot case at 0.72 AU (near-Venus), and 2) a cold case at 5.0 AU (near-Jupiter). Fig. 3 indicates the direction of the solar flux. The central tower region was of great interest thermally, and therefore, an enclosure around this area was constructed in the GMM so that a global verification of the form factor calculation could be obtained (see Fig. 4). Initially, form factors were computed by using the standard TRASYS values in the FFCAL segment. The nodal form factor sums for some of the central tower nodes are summarized in Table 2, along with the corresponding form factors to space and absorbed solar heating at 1 AU. Afterward, the form factor matrix was normalized. The enclosing the open geometry resulted in 20 GMM nodes outside of the acceptable form factor sum range between 0.95 and 1.05. These nodes were recomputed using the Nusselt unit sphere technique,

Table 1 - GMM Exterior Node Description

Node Number(s)	Description	Exterior Surface	α_p/ϵ
1 - 4	Bus shade, HGA side	Black Kapton	0.85/0.75
5 - 11	Lower Tower	Black Kapton	0.85/0.75
12	Radome	Black Kapton	0.85/0.75
68, 69	Upper Support Structure	Black Kapton	0.85/0.75
71 - 75	PWS Support Structure	Black Kapton	0.85/0.75
76, 77	Tip Shade Support Structure	Black Paint	0.93 /0.87
78, 79	Tip Shade	Carbon-Filled Kapton	0.90/0.81
80	LGA	White Paint	0.30/0.85
81, 82	Tip Shield MLI	ITO-Coated Carbon-Filled Kapton	0.50/0.71
113 - 118 150 -153	Rib MLI	Black Kapton	0.85/0.75

Table 2 - Summary of Normalization Process

OPEN GEOMETRY				FORM FACTOR MATRIX NORMALIZATION				
Node No.	Form Factor Sum	Form Factor to Space	Absorbed Solar Heating at 1 AU, W	Open Geometry with Enclosure	After Recomp	After Reduction & Set to Unity	Form Factor to Space*	Absorbed Solar Heating at 1 AU, W
1	0.1274	0.8726	5266.5	0.9803	0.9771	1.0000	0.8529 (<)	5282.5
4	0.3257	0.6743	8.810	1.0215	1.0357	1.0033	0.6684 (<)	8.709
7	0.4514	0.5486	0.676	0.9673	0.9672	1.0000	0.5159 (<)	0.661
10	0.5779	0.4231	10.55	0.9007	1.0093	1.0000	0.3390 (<)	9.321
12	0.6009	0.3991	10.95	1.0600	1.0100	1.0000	0.4339 (>)	9.808
69	0.7797	0.2203	0.643	1.1016	1.0066	0.7817	0.2183 (<)	0.611
71	0.7869	0.2131	0.549	0.9314	0.9672	1.0000	0.1852 (<)	0.549
72	0.7273	0.2727	0.499	0.9333	1.0322	1.0112	0.2787 (>)	0.360
76	0.6272	0.3728	0.000	1.0508	0.9975	1.0000	0.3927 (>)	0.000

Note:

"<" indicates the value is less than open geometry form factor to space, and ">" indicates the value is greater than the open geometry form factor to space

and following this, all nodal form factor sum were within an acceptable range. Next, the formfactor sum that exceed unity are reduced and then, all the form factor sums are adjusted to unity. Lastly, the enclosing nodes are removed, and the adjusted form factor matrix for the open geometry remains. Radiation conductors and absorbed heating were calculated. Table 2 summarizes the normalization process. Once radiation conductors and absorbed heating were determined, temperature estimates were determined at 0.72 AU and 5.0 AU using the thermal model from Ref. 6, and the results are given in Table 3.

Discussion of Results

A quick glance at the temperature results indicates that the difference between the standard form factor calculation and form factor matrix normalization may be as larger as 7°C in the hot case and 3°C in the cold case. For the hot case, notice that the temperature of node 7 is warmer for form factor normalization when compared with the standard calculation. However, it should be also indicated that the temperature of node 72 is cooler when the same comparison is made. There is appears to be no apparent trend when comparing temperature differences. However, when Table 2 is reviewed for the comparison between the form factor to space, a pattern develops. In general, when the form factor to space using form factor normalization is less than that of the standard calculation the temperature using the normalization method is greater than the corresponding temperature using the standard technique. In addition, the converse appears to be generally true. A reduced form factor to space usually implies a warmer nodal temperature. However, node 10 is an exception to this generalization, and it seems more influenced by the part of the normalization process where form factors are recomputed to obtain a nodal form factor sum between 0.95 and 1.05. The initial form factor sum within the enclosure was 0.9007 and after recomputation, it was increased to 1.0093. Consequently, this may have changed not only the form factor to space, but also other internodal form factors may have increased or decreased. The normalization process does not always reduce the form factor to space, but rather, it attempts to distribute the form factor computational error over all the nodes. In the process, the analyst strives to verify and revise the form factor calculation in a global way.

The temperature differences in the cold case are less marked than the hot case. At 5.0 AU, the environmental heat load is much smaller than at 0.72 AU, and the temperature distribution should be driven by the radiation coupling to space. For the most part, the form factor to space between the two methods are small, thus leading to only small temperature differences.

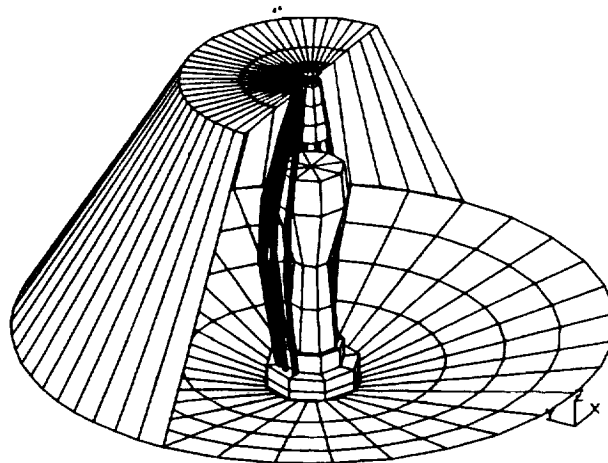


Figure 4 - Enclosing geometry for HGA GMM. Portions of the tip shade, ribs, and close-out removed for clarity.

Table 3 - Hot and Cold Case Temperature Estimates in °C

Node No.	Open Geometry		Form Factor Matrix Normalization		$\Delta T = T_{\text{norm}} - T_{\text{open}}^a$, °C	
	Hot	Cold	Hot	Cold	Hot	Cold
1	-3.2	-168.4	-2.2	-167.9	1.0	0.5
4	-3.2	-168.4	-2.2	-167.9	1.0	0.5
7	6.0	-152.2	10.1	-150.4	4.1	1.8
10	68.1	-137.7	65.9	-138.3	-2.2	-0.6
12	70.0	-141.6	634.1	-143.3	-5.9	-2.0
69	64.0	-143.2	63.7	-143.3	-0.3	-0.1
71	70.3	-141.4	72.2	-140.6	2.2	0.8
72	69.0	-99.6	62.2	-100.1	-6.8	-0.5
76	72.2	-133.2	68.9	-136.1	-3.3	-2.9
33 ^b	35.7	-38.8	35.8	-38.7	0.1	0.1
65 ^c	63.9	-145.0	59.4	-146.2	-4.5	-1.2

Notes:

- ^a Temperature difference between form factor normalization and standard (open geometry) approaches
- ^b Internal node - DDA
- ^c Internal node - S-band antenna feed

Two internal thermal model nodes have been included in Table 3. The DDA (node 33) is coupled to a 25°C boundary, and is largely unaffected by normalization. However, the S-band antenna feed (node 65) is more responsive to the external radiative environment, and this environment can be characterized by node 12 (see Fig. 3). Since the temperature of node 12 for normalization is cooler than the standard method, the S-band antenna feed has a similar character.

When dealing with thermal models, the question of uncertainty arises frequently. As inferred from the results of this sample case, unverified form factor calculations may cause an uncertainty of approximately $\pm 5^\circ\text{C}$. Unless the thermal design is very forgiving, unverified form factors could result in optimistic thermal performance. Therefore, some method of form factor validation should be performed, and form factor normalization provides such an avenue.

CONCLUSIONS

A method that may globally verify and revise TRASYS form factor calculations has been presented. The primary features of this approach are reducing form factors on a weighed form factor basis and adding a self-viewing form factor to adjust nodal form factor sums to unity. In comparison to the standard method of determining form factors, this process may result in temperatures that may differ by $\pm 5^{\circ}\text{C}$. It is recommended that this approach be utilized so that form factor computational error would be distributed over the entire geometric model rather than any one node.

ACKNOWLEDGEMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration. The author would like to thank Ruben Rivera of Rockwell International for providing the PL-PULL program and documentation. Duane Beach of NASA Lewis provided extensive documentation regarding various pitfalls with TRASYS. Appreciation is also due to Bob Wise and Laura Mathiowetz for explaining their direct form factor to space calculation difficulties.

REFERENCES

1. *Thermal Radiation Analyzer System (TRASYS) User's Manual*, Johnson Space Center, Houston, Texas, December 1987
2. Wise, R. "Things to Avoid When Using TRASYS," *Jet Propulsion Laboratory Internal Document*, Pasadena, California, December 20, 1991.
3. Grondalski, L. "Comparison of MacTRASYS and MacSINDA Calculations to VAX TRASYS and SINDA Calculations for the PMS Radiation Heat Pipe Model," *Jet Propulsion Internal Document 3548-CAS-92-042*, Pasadena, California, March 10, 1992.
4. Richmond, M. "TRASYS," *Goddard Space Flight Center Internal Document*, Greenbelt, Maryland, May 1988.
5. Weatherford, R. "Computer Program PL-PULL," *Rockwell International Internal Document SAS/TA-PTI-79-029*, Downey, California, July 2, 1979.
6. Greenfield, M. "Thermal Analyses and Results of Studies Performed on the Galileo High Gain Antenna for the VEEGA Mission," *Jet Propulsion Laboratory Internal Document 354-GLL-88-025*, Pasadena, California, May 2, 1988.

N93713392

A SIMPLE NODE AND CONDUCTOR DATA GENERATOR FOR SINDA

Ronald R. Gottula
GENCORP Aerojet
Azusa, California

SUMMARY

This paper presents a simple, automated method to generate NODE and CONDUCTOR DATA for thermal math models. The method uses personal computer spreadsheets to create SINDA inputs. It was developed in order to make SINDA modeling less time consuming and serves as an alternative to graphical methods.

Anyone having some experience using a personal computer can easily implement this process. The user develops spreadsheets to automatically calculate capacitances and conductances based on material properties and dimensional data. The necessary node and conductor information is then taken from the spreadsheets and automatically arranged into the proper format, ready for insertion directly into the SINDA model.

This technique provides a number of benefits to the SINDA user such as a reduction in the number of hand calculations, and an ability to very quickly generate a parametric set of NODE and CONDUCTOR DATA blocks. It also provides advantages over graphical thermal modeling systems by retaining the analyst's complete visibility into the thermal network, and by permitting user comments anywhere within the DATA blocks.

INTRODUCTION

There continues to be a need to more fully automate thermal modeling. As a member of the engineering team, the thermal analyst is being asked to perform more comprehensive studies on more complex systems, and to do so in less time.

Over the years, a number of techniques have been developed to make thermal modeling more productive. For instance, various computer programs have been written for transforming finite element models into "equivalent" finite difference models (refs. 1, 2, 3, 4, 5). Other systems avoid finite differences altogether and solve for temperatures directly using the finite element formulation (refs. 6, 7, 8, 9, 10).

However, many thermal analysts do not have access to or training on these graphical systems. Recognizing these constraints, a new thermal network generator utilizing a personal computer spreadsheet program was devised.

IMPLEMENTATION

All that is needed to implement this method is a personal computer, a spreadsheet program, a programming language compiler, and a way to transfer files from the PC to the computer used to run SINDA.

NODE DATA GENERATOR

The first step is to construct a NODE DATA spreadsheet, similar to the one in Figure 1, which calculates nodal capacitances. One node is defined on each row of the spreadsheet. Each column contains a different type of information about the node, such as the node number, initial temperature, material properties, and dimensions. The cell in the next-to-last column contains an equation which calculates the nodal capacitance as a function of the material properties and dimensions. The last column contains a comment to be added at the end of that NODE DATA record.

Several rows in the spreadsheet begin with "C" or "C ***". These will serve as comment lines in the finished NODE DATA block.

Some of the cells in the "Area" column have a "-". For these nodes, the volume calculation is a product of the thickness, width and length.

Some of the cells in the "Thickness" and "Width" columns have a "-". For these nodes, the volume calculation is a product of the cross-sectional area and length.

In order to remain consistent with SINDA NODE DATA input codes, arithmetic nodes are defined by entering a capacitance value of "-1.0". Similarly, boundary and heater nodes are defined by entering negative node numbers.

Once the spreadsheet is completed, it should be written to disk (saved) as usual. But before exiting the spreadsheet program, the information needs also to be saved in ASCII form so that it can eventually be taken to the SINDA model. In Microsoft Excel, for instance, the "Save As..." command is used, and the CSV (comma separated values) file format option is chosen. This produces a file similar to the one shown in Figure 2.

The CSV file contains some data that is not useful (nor legal) in the SINDA NODE DATA block. Therefore, the user must write a simple FORTRAN

or BASIC program which will read in the CSV file and write out the NODE DATA in proper SINDA format. A sample BASIC program which does this, NODE.BAS, is shown in Figure 3.

NODE.BAS reads in the CSV file one line at a time. All lines before "C" or "C ***" is encountered are ignored. Any lines beginning with "C" or "C ***" are written out as comment lines. All of the other lines contain actual NODE DATA. These are read in one value at a time, with the pertinent values (node number, initial temperature, capacitance, comment) written out in proper SINDA format. The resulting file, shown in Figure 4, is ready to be inserted directly into the SINDA model.

CONDUCTOR DATA GENERATOR

The CONDUCTOR DATA generator is very similar to the NODE DATA generator. An example is shown in Figures 5 through 8.

Some of the "Thickness" values in the CONDUCTOR DATA spreadsheet (Figure 5) are entered as "-". In these cases, the "Conductance" value is calculated using a cross-sectional area rather than a product of the thickness and width.

This spreadsheet can be used to consolidate a series of conductances into a single SINDA conductor. For example, conductor 19 in Figure 5 is a case where two conductances (one through a solid, the next through a joint) are consolidated into a single SINDA conductor. A simple KA/L is calculated in the "Conductance" column. A "Joint Conductance" is then calculated as a function of the number of fasteners in the joint. Finally, these two conductances are taken in series to calculate an equivalent "Final C'tance."

DISCUSSION

Using spreadsheets to generate NODE and CONDUCTOR DATA provides a number of advantages compared to traditional (non-automated) and graphical methods.

This method is perhaps as similar to traditional SINDA modeling as any automated method can be. The spreadsheets look very much like the actual NODE and CONDUCTOR DATA blocks, including "-1.0" capacitances for arithmetic and heater nodes, negative node numbers for boundary and heater nodes, "dollar sign" comments at the ends of records, and a "C" in column one for comment lines. Because of these similarities, practically any SINDA user can quickly understand and use the spreadsheets.

One productivity advantage is the user's ability to quickly edit material properties and dimensions. As any spreadsheet user knows, entire columns of entries can be edited very quickly. Parametric analysis becomes convenient by rapidly editing and saving an assortment of NODE and CONDUCTOR DATA blocks representing a variety of design alternatives. Once the node and conductor files are placed into the same directory as the SINDA model, the SINDA '85 INCLUDE statement can be used to automatically insert these external files into the SINDA model as appropriate.

Another productivity advantage is gained because of the virtual elimination of hand calculations. Once the spreadsheets have been developed and checked for accuracy, capacitance and conductance updates become immediate and remain accurate as design changes are incorporated.

A great deal of flexibility is possible using the spreadsheets. GEN, SIV and other SINDA NODE and CONDUCTOR DATA options may be incorporated. Parameters of any type may be included. Very complex equations can be defined. For example, a complicated joint conductance equation may be entered as a function of parameters such as clamping force, surface roughness, etc.

Another benefit of the spreadsheet method is that the thermal analyst, as well as his supervisor and the customer, continue to have full visibility into the thermal model network. Most graphical SINDA modelers make the computations and assumptions behind the capacitance and conductance values (and sometimes even the values themselves) transparent to the user. Also, they often use finite element to finite difference translations which create "cross conductors" or negative conductors. The resulting thermal networks can be difficult for some users to understand and scrutinize for validity.

Unlike most graphical modelers, the spreadsheet method permits a user to fully document the NODE and CONDUCTOR DATA since user comments may be inserted anywhere in the blocks.

The spreadsheet method allows a good paper trail to be maintained. Archived spreadsheet files and printouts can be kept to retain historical information. Most spreadsheet programs have an automatic date/time stamp capability. For example, the Microsoft Excel function "=NOW()" will read the date and time from the PC's internal clock and display them in a cell, as shown in Figures 1 and 5.

CONCLUSION

An automated method of generating NODE and CONDUCTOR DATA has been developed utilizing personal computer spreadsheets. The spreadsheet method has a "look and feel" familiar to SINDA users, provides relief

from hand calculations, allows flexibility in capacitance and conductance formula definitions, retains visibility into the thermal network, and permits user comments anywhere in the DATA blocks.

The spreadsheet method is an attractive alternative to traditional SINDA modeling for those thermal analysts who cannot take advantage of graphical modelers.

REFERENCES

1. Anderson, T./Baum, F./Zarda, P.R., Thermal Analysis with Supertab and FEM/SINDA, SDRC 1988 I-DEAS/CAEDS Users' Conference Series, International Proceedings, SDRC, 2000 Eastman Drive, Milford, OH 45150-2789
2. SDRC I-DEAS TMG (Thermal Model Generator) User's Guide, Version 6.0, December 1991, MAYA Heat Transfer Technologies Ltd., 43 Thornhill, Montreal, P.Q., Canada H3Y 2E3
3. P/Thermal 2.3 User Manual, July 1988, PDA Engineering, PATRAN Division, 2975 Redhill Avenue, Costa Mesa, CA 92626
4. PAT/SINDA Application Interface Guide, Release 2.1A, March 1987, PDA Engineering, PATRAN Division, 2975 Redhill Avenue, Costa Mesa, CA 92626
5. ANSYS-SINDA/1987 Interface, Network Analysis Associates, Inc., 9972 Sage Circle, Fountain Valley, CA 92708
6. SDRC I-DEAS Model Solution User's Guide, Level 6.0, December 1990, SDRC, 2000 Eastman Drive, Milford, OH 45150-2789
7. P/FEA 2.3 User Manual, July 1988, PDA Engineering, PATRAN Division, 2975 Redhill Avenue, Costa Mesa, CA 92626
8. DeSalvo, G.J./Gorman, R.W., ANSYS Engineering Analysis System User's Manual, May 1, 1989, Swanson Analysis Systems, Inc., P. O. Box 65, Houston, PA 15342
9. Harder, R.L., MSC/NASTRAN Thermal Analysis, April 1986, MacNeal-Schwendler Corp., 815 Colorado Blvd., Los Angeles, CA 90041-1777
10. PCB Thermal User's Reference Manual, Version 2.2, 05 October 1990, Pacific Numerics Corporation, La Jolla, CA

NODE DATA for SINDA model 'SAMPLE.INP'									
This is spreadsheet file 'SAMPLE N.XLS'									
Node	Init'l	Mat'l	Mat'l						
Number	Temp	Density	Spec Ht	Thcknss	Width	Area	Length	Capacitance	Comments
C									
C *** THE FOLLOWING NODES WERE GENERATED BY 'SAMPLE N.XLS'									
C *** ON 12 June 92, 2:16 PM									
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92									
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP									
C									
C *** STRUCTURE NODES									
C *** STRUCTURE TORQUE BOX									
102	30.0	0.101	0.109	0.070	6.290	-	8.170	0.0396	\$
105	30.0	0.101	0.109	0.070	6.290	-	8.170	0.0396	\$
108	30.0	0.101	0.109	0.070	4.160	-	8.170	0.0262	\$
109	30.0	0.101	0.109	0.070	4.160	-	8.170	0.0262	\$
112	30.0	0.101	0.109	0.070	7.900	-	8.170	0.0497	\$
152	30.0	0.101	0.109	0.070	6.290	-	7.970	0.0386	\$
155	30.0	0.101	0.109	0.070	6.290	-	7.970	0.0386	\$
158	30.0	0.101	0.109	0.070	4.160	-	7.970	0.0256	\$
159	30.0	0.101	0.109	0.070	4.160	-	7.970	0.0256	\$
162	30.0	0.101	0.109	0.070	7.900	-	7.970	0.0485	\$
C									
C *** STRUCTURE WALL TRUSSES									
104	30.0	0.101	0.109	-	-	0.090	8.214	-1.0	\$
204	30.0	0.101	0.109	-	-	0.090	4.215	-1.0	\$
154	30.0	0.101	0.109	-	-	0.090	16.547	0.0164	\$
361	30.0	0.101	0.109	-	-	0.090	9.300	-1.0	\$
385	30.0	0.101	0.109	-	-	0.090	9.300	-1.0	\$
C									
C									

Figure 1. NODE DATA Spreadsheet

```

NODE DATA for SINDA model 'SAMPLE.INP' .....
This is spreadsheet file 'SAMPLE_N.XLS' .....

.....
Node,Init'l,Mat'l,Mat'l,.....
Number,Temp,Density,Spec Ht,Thcknss,Width,Area,Length,Capacitance,Comments
.....
C.....
C *** THE FOLLOWING NODES WERE GENERATED BY 'SAMPLE_N.XLS' .....
C *** ON 12 June 92, 2:49 PM,.....
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92,.....
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP,.....
C.....
C *** STRUCTURE NODES,.....
C *** STRUCTURE TORQUE BOX,.....
102,30.0,0.101,0.109,0.070,6.290,-,8.170,0.0396,$
105,30.0,0.101,0.109,0.070,6.290,-,8.170,0.0396,$
108,30.0,0.101,0.109,0.070,4.160,-,8.170,0.0262,$
109,30.0,0.101,0.109,0.070,4.160,-,8.170,0.0262,$
112,30.0,0.101,0.109,0.070,7.900,-,8.170,0.0497,$
152,30.0,0.101,0.109,0.070,6.290,-,7.970,0.0386,$
155,30.0,0.101,0.109,0.070,6.290,-,7.970,0.0386,$
158,30.0,0.101,0.109,0.070,4.160,-,7.970,0.0256,$
159,30.0,0.101,0.109,0.070,4.160,-,7.970,0.0256,$
162,30.0,0.101,0.109,0.070,7.900,-,7.970,0.0485,$
C.....
C *** STRUCTURE WALL TRUSSES,.....
104,30.0,0.101,0.109,-,0.090,8.214,-1.0,$
204,30.0,0.101,0.109,-,0.090,4.215,-1.0,$
154,30.0,0.101,0.109,-,0.090,16.547,0.0164,$
361,30.0,0.101,0.109,-,0.090,9.300,-1.0,$
385,30.0,0.101,0.109,-,0.090,9.300,-1.0,$
C.....
C

```

Figure 2. NODE DATA CSV File

' NODE.BAS Compiled using Borland TurboBASIC

```
Y = INSTR(COMMAND$, " ")
INFILE$ = LEFT$(COMMAND$, Y-1)
OUTFILE$ = RIGHT$(COMMAND$, LEN(COMMAND$)-Y)
OPEN INFILE$ FOR INPUT AS #1
OPEN OUTFILE$ FOR OUTPUT AS #2

20 INPUT #1, A$
IF EOF(1) THEN STOP
IF A$ = "C" OR LEFT$(A$, 5) = "C ****" THEN
    PRINT #2, A$
    GOTO 30
ELSE
    GOTO 20
END IF

30 INPUT #1, A$
IF EOF(1) THEN STOP
IF LEN(A$) = 0 THEN GOTO 30
IF A$ = "C" OR LEFT$(A$, 5) = "C ****" THEN
    PRINT #2, A$
    GOTO 30
END IF
INPUT #1, B$, C$, D$, E$, F$, G$, H$, I$, J$
IF EOF(1) THEN STOP
A$ = STRING$(14-LEN(A$), " ") + A$ + ","
B = INT(VAL(B$))
B1$ = STR$(B)
IF LEN(B$) - LEN(B1$) < 2 THEN
    B$ = STRING$(12-LEN(B1$), " ") + B1$ + ".0,"
ELSE
    B$ = STRING$(12-LEN(B1$), " ") + B$ + ","
END IF
I = INT(VAL(I$))
I1$ = STR$(I)
IF I > 9 THEN
    I$ = I$ + STRING$(27-(LEN(I$)-LEN(I1$)-1), " ")
ELSE
    I$ = " " + I$ + STRING$(27-(LEN(I$)-LEN(I1$)-1), " ")
END IF
IF LEFT$(J$, 1) = "$" THEN
    J$ = LEFT$(J$, 12)
ELSE
    J$ = "$ " + LEFT$(J$, 10)
END IF
PRINT #2, A$; B$; I$; J$

GOTO 30
STOP
```

Figure 3. NODE.BAS

```

C
C *** THE FOLLOWING NODES WERE GENERATED BY 'SAMPLE_N.XLS'
C *** ON 12 June 92, 2:49 PM
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP
C
C *** STRUCTURE NODES
C *** STRUCTURE TORQUE BOX
      102,      30.0, 0.0396      $
      105,      30.0, 0.0396      $
      108,      30.0, 0.0262      $
      109,      30.0, 0.0262      $
      112,      30.0, 0.0497      $
      152,      30.0, 0.0386      $
      155,      30.0, 0.0386      $
      158,      30.0, 0.0256      $
      159,      30.0, 0.0256      $
      162,      30.0, 0.0485      $
C
C *** STRUCTURE WALL TRUSSES
      104,      30.0, -1.0      $
      204,      30.0, -1.0      $
      154,      30.0, 0.0164      $
      361,      30.0, -1.0      $
      385,      30.0, -1.0      $
C
C

```

Figure 4. Final NODE DATA Block

CONDUCTOR DATA for SINDA model 'SAMPLE.INP'											
This is spreadsheet file 'SAMPLE.C.XLS'											
Conductor Number	Node i	Node j	Conductivity	Thickness	Width	Length	Conductance	Number Fasteners	Joint C'tance	Final C'tance	Comments
C *** THE FOLLOWING CONDUCTORS WERE GENERATED BY 'SAMPLE.C.XLS'											
C *** ON 15 June 92, 3:16 PM											
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92											
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP											
C *** STRUCTURE CONDUCTORS											
C *** STRUCTURE TORQUE BOX											
1	102	105	3.96	0.070	6.180	4.070	0.421	-	-	0.421	\$
3	102	108	3.96	0.070	6.180	4.070	0.421	-	-	0.421	\$
5	102	109	3.96	0.070	8.140	3.090	0.730	-	-	0.730	\$
7	102	112	3.96	0.070	8.140	3.090	0.730	-	-	0.730	\$
9	112	152	3.96	0.070	6.180	4.070	0.421	-	-	0.421	\$
11	112	155	3.96	0.070	6.180	4.070	0.421	-	-	0.421	\$
13	112	158	3.96	0.070	8.140	3.090	0.730	-	-	0.730	\$
15	112	159	3.96	0.070	8.140	3.090	0.730	-	-	0.730	\$
17	159	162	3.96	0.070	4.110	4.070	0.280	-	-	0.280	\$
19	162	102	3.96	0.070	4.110	4.070	0.280	2	0.5	0.179	\$
C *** STRUCTURE WALL TRUSSES											
51	104	204	3.96	-	0.490	4.360	0.445	-	-	0.445	\$
53	204	154	3.96	-	0.490	4.360	0.445	-	-	0.445	\$
55	154	361	3.96	-	0.490	4.150	0.468	-	-	0.468	\$
57	361	385	3.96	-	0.090	4.880	0.073	-	-	0.073	\$
59	385	104	3.96	-	0.090	4.880	0.073	-	-	0.073	\$
C											
C											

Figure 5. CONDUCTOR DATA Spreadsheet

```

CONDUCTOR DATA for SINDA model 'SAMPLE.INP' ,,,,,,,,,
This is spreadsheet file 'SAMPLE_C.XLS' ,,,,,,,,,
,,,,,,,,
Cond,Node,Node,Cond-,Thick-,,,,Conduc-,Number,Joint,Final,
Number,i,j,tivity,ness,Width,Length,tance,Fasteners,C'tance,C'tance,Comments
,,,,,,,,
C,,,,,,,,
C *** THE FOLLOWING CONDUCTORS WERE GENERATED BY 'SAMPLE_C.XLS' ,,,,,,,,,
C *** ON 15 June 92, 3:16 PM, ,,,,,,,,,
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92, ,,,,,,,,,
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP, ,,,,,,,,,
C,,,,,,,,
C *** STRUCTURE CONDUCTORS, ,,,,,,,,,
C *** STRUCTURE TORQUE BOX, ,,,,,,,,,
1,102,105,3.96,0.070,6.180,4.070,0.421,-,-,0.421,$
3,102,108,3.96,0.070,6.180,4.070,0.421,-,-,0.421,$
5,102,109,3.96,0.070,8.140,3.090,0.730,-,-,0.730,$
7,102,112,3.96,0.070,8.140,3.090,0.730,-,-,0.730,$
9,112,152,3.96,0.070,6.180,4.070,0.421,-,-,0.421,$
11,112,155,3.96,0.070,6.180,4.070,0.421,-,-,0.421,$
13,112,158,3.96,0.070,8.140,3.090,0.730,-,-,0.730,$
15,112,159,3.96,0.070,8.140,3.090,0.730,-,-,0.730,$
17,159,162,3.96,0.070,4.110,4.070,0.280,-,-,0.280,$
19,162,102,3.96,0.070,4.110,4.070,0.280,2.05,0.179,$
C *** STRUCTURE WALL TRUSSES, ,,,,,,,,,
51,104,204,3.96,-,0.490,4.360,0.445,-,-,0.445,$
53,204,154,3.96,-,0.490,4.360,0.445,-,-,0.445,$
55,154,361,3.96,-,0.490,4.150,0.468,-,-,0.468,$
57,361,385,3.96,-,0.090,4.880,0.073,-,-,0.073,$
59,385,104,3.96,-,0.090,4.880,0.073,-,-,0.073,$
C,,,,,,,,
C,,,,,,,,

```

Figure 6. CONDUCTOR DATA CSV File

' CONDUCT.BAS Compiled using Borland TurboBASIC

```
Y = INSTR(COMMAND$, " ")
INFILE$ = LEFT$(COMMAND$, Y-1)
OUTFILE$ = RIGHT$(COMMAND$, LEN(COMMAND$)-Y)
```

```
OPEN INFILE$ FOR INPUT AS #1
OPEN OUTFILE$ FOR OUTPUT AS #2
```

```
20 INPUT #1, A$
IF EOF(1) THEN STOP
IF A$ = "C" OR LEFT$(A$,5) = "C ***" THEN
  PRINT #2, A$
  GOTO 30
ELSE
  GOTO 20
END IF
```

```
30 INPUT #1, A$
IF EOF(1) THEN STOP
IF LEN(A$) = 0 THEN GOTO 30

IF A$ = "C" OR LEFT$(A$,5) = "C ***" THEN
  PRINT #2, A$
  GOTO 30
END IF
```

```
INPUT #1, B$, C$, D$, E$, F$, G$, H$, I$, J$, K$, L$
IF EOF(1) THEN STOP
```

```
A$ = STRING$(14-LEN(A$), " ") + A$ + ", "
```

```
B$ = STRING$(11-LEN(B$), " ") + B$ + ", "
```

```
C$ = STRING$(7-LEN(C$), " ") + C$ + ", "
```

```
K = INT(VAL(K$))
```

```
K1$ = STR$(K)
```

```
K$ = STRING$(7-LEN(K1$), " ") + K$ + STRING$(18-(LEN(K$)-LEN(K1$)-1), " ")
```

```
IF LEFT$(L$,1) = "$" THEN
```

```
  L$ = LEFT$(L$,12)
```

```
ELSE
```

```
  L$ = "$ " + LEFT$(L$,10)
```

```
END IF
```

```
PRINT #2, A$; B$; C$; K$; L$
```

```
GOTO 30
```

```
STOP
```

Figure 7. CONDUCT.BAS

```

C
C *** THE FOLLOWING CONDUCTORS WERE GENERATED BY 'SAMPLE_C.XLS'
C *** ON 15 June 92, 3:16 PM
C *** DIMENSIONS ARE FROM CAD MODEL 'STRUCTURE.MF1;14' 04 JUNE 92
C *** ALUMINUM 2014-T6 PROPERTIES ARE FROM MIL-HDBK-5F, ROOM TEMP
C
C *** STRUCTURE CONDUCTORS
C *** STRUCTURE TORQUE BOX
      1,      102,      105,      0.421      $
      3,      102,      108,      0.421      $
      5,      102,      109,      0.730      $
      7,      102,      112,      0.730      $
      9,      112,      152,      0.421      $
     11,      112,      155,      0.421      $
     13,      112,      158,      0.730      $
     15,      112,      159,      0.730      $
     17,      159,      162,      0.280      $
     19,      162,      102,      0.179      $
C *** STRUCTURE WALL TRUSSES
     51,      104,      204,      0.445      $
     53,      204,      154,      0.445      $
     55,      154,      361,      0.468      $
     57,      361,      385,      0.073      $
     59,      385,      104,      0.073      $
C
C

```

Figure 8. Final CONDUCTOR DATA Block

Development Status of SINDA/FLUINT and SINAPS

Brent A. Cullimore and Steven G. Ring
Martin Marietta Astronautics Group
Denver, Colorado

Eugene K. Ungar
NASA Johnson Space Center
Houston, Texas

Summary

SINDA/FLUINT (Systems Improved Numerical Differencing Analyzer / Fluid Integrator, formerly SINDA '85) is a computer code used to analyze thermal/fluid systems that can be represented in lumped parameter form. In addition to conduction and radiation heat transfer, the code is capable of modeling both single- and two-phase flow networks, their associated hardware, and their heat transfer processes. In this paper, recent improvements to SINDA/FLUINT are described, as are those in progress that will be available in the fall of 1992 in Version 2.5. Also, a preview of planned enhancements is provided. This paper also introduces SINAPS (SINDA Application Programming System), a powerful graphical pre- and postprocessor that will also be available in the fall of 1992.

Background

Evolving spacecraft thermal control technology is increasingly utilizing two-phase fluid systems to accomplish waste heat acquisition, transport, and rejection. In the case of the Space Station Freedom, the high heat rejection requirement of 82.2 kW and the typical heat transport distances of over 100 feet made a two-phase thermal control system the only rational choice. A conventional heat pipe or single-phase fluid loop thermal control system, such as have been used in previous US spacecraft, would have had unacceptable weight and power penalties. The heat rejection requirements will be even higher and the transport distances will be even longer for lunar and planetary base applications, again forcing the use of two-phase thermal control systems for those missions.

The introduction of two-phase active thermal control systems required a quantum leap in the development of thermal control technology. A similar development effort was required for the analytical tools for modeling such systems. Previously, there was no single computer tool that was suitable for analyzing spacecraft two-phase systems and components, especially when the requirement was levied to integrate such analyses with vehicle-level simulation tools such as SINDA and TRASYS (Thermal Radiation Analysis System). Typically, two-phase systems and components were analyzed by generating application-unique mathemati-

cal modeling equations that were then incorporated into numerical solution computer programs. This method of analysis caused much duplication of effort and hindered the transfer of thermal math models and their ability to be modified by other analysts.

Therefore, in the mid 1980's NASA Johnson Space Center (JSC) launched an effort to develop a design simulation tool that was well suited to modeling two-phase systems for space applications. An effort was already nearing completion at NASA JSC which brought the 1972 version of SINDA up to modern standards, completely reworking it and adding submodels and other capabilities that enhance model integration and exchange. The result of that modernization, called SINDA '85, was used as a starting point for the addition of the new fluid analysis capabilities. The final product, SINDA/FLUINT, is a quantum leap above the older versions of SINDA, featuring a comprehensive single- and two-phase, steady and transient fluid analysis package (FLUINT) that works together with traditional SINDA thermal networks to solve arbitrarily complex thermal/fluid problems. Version 2.3, released in early 1990, has become the most commonly used tool for analysis of fluid flow and heat transfer in space-based systems, and has spread to other specialties (propulsion, environmental control) and even other industries (energy, aircraft, automotive, and architectural) because of its generality, analytic power, transportability, and ability to be customized. In 1991, SINDA/FLUINT was awarded the NASA Space Act Award.

SINDA/FLUINT has been continually updated and enhanced since its first release in the late 1980's. The improvements have made the code even more general in scope, better able to handle different and more difficult problems, and more efficient in its use of computer time. References 1, 2, and 3 describe the capabilities of Version 2.3, which is available through COSMIC, NASA's software distributor. In this paper, the capabilities of the current NASA version (Version 2.4) are described, as well as the work currently being completed by Martin Marietta on Version 2.5, which will be available in the fall of 1992. Improvements planned for future versions are also described.

In 1990, NASA JSC initiated an effort to provide a modern graphical pre- and postprocessor for SINDA/FLUINT. Martin Marietta is currently completing the result of this effort: SINAPS, a powerful graphical interface that will be available in the fall of 1992. SINAPS provides a means for graphically building and maintaining SINDA/FLUINT models, and displaying the results on the sketch the user has created. In this paper, the capabilities of SINAPS are detailed.

SINDA/FLUINT Enhancements

Almost all work since 1985 has focused on the continuing development and expansion of the fluid analysis capabilities; only relatively minor improvements have been made to the thermal analysis code. This section lists and describes the major advances in the FLUINT portion of the code.

There has been a steady accrual of relatively minor expansions and corrections over the years. While collectively these improvements have added significantly to the speed, utility, and ruggedness of the code, they are too numerous and detailed to be described in this paper. Suffice it to say that few users regret the effort required to update their version even if the latest round of major improvements was not directly of interest to them.

Version 2.4 Enhancements

The primary goal of Version 2.4, completed in December 1991 and documented in Reference 4, was to enable the user to selectively avoid the assumption of homogeneous two-phase flow, and to use instead a slip flow formulation. To achieve this goal, various important features had to be added to the code in preparation for the addition of slip flow modeling, such as flow regime mapping. The ability to discern basic flow regimes and to calculate the frictional pressure drop accordingly can be used independently of the slip flow options. Flow regime mapping options are described first, followed by slip flow modeling options.

Flow Regime Mapping Options—Packaged as an optional pressure drop ‘correlation,’ the user may elect to have a simplified two-phase flow regime predicted for duct segments, with the pressure gradient estimated on the basis of that regime. Output routines have been modified to print the current flow regime if this option is used and the flow is two-phase. Instead, if the flow is single-phase, the Reynolds number is printed instead.

Four generalized (simplified) regimes are recognized, as illustrated in Figure 1: bubbly, slug, annular, and stratified. The first two are considered ‘dispersed,’ and the latter two ‘separated.’ The distinction between regimes is based (1) on the liquid and vapor mass fluxes, (2) on the void fraction, (3) on the hydraulic diameter of the line—assumed nearly circular, (4) on the magnitude of a body force (or acceleration) vector and its orientation with respect to the duct, (5) on fluid properties such as densities, viscosities, and surface tension, and (6) in the event no clear determination can be made, on previous flow regimes (i.e., regime boundaries exhibit hysteresis). Flow regime mapping methods identified in Reference 5 were used extensively, although neither exactly nor exclusively.

Bubbly flow occurs at the extremes of low gravities, high liquid mass fluxes compared to the vapor flux, and low void fractions (less than about 0.46), and is characterized by small vapor bubbles entrained in liquid. If the bubbles coalesce due to increased accelerations, decreased liquid mass flux, or increased void fraction, then the slug flow regime will appear. The slug flow regime exhibits large bubbles that nearly span the diameter of the tube, but which are axially separated from each other by liquid. Both the slug and bubbly flow regimes are characterized by relatively little slip flow, approaching true homogeneous flow. In both cases, predicted pressure drops are based on the McAdam’s formulation for homogeneous flow. These two regimes are therefore identical for homogeneous passages, but they behave differently if slip flow is modeled, as described later.

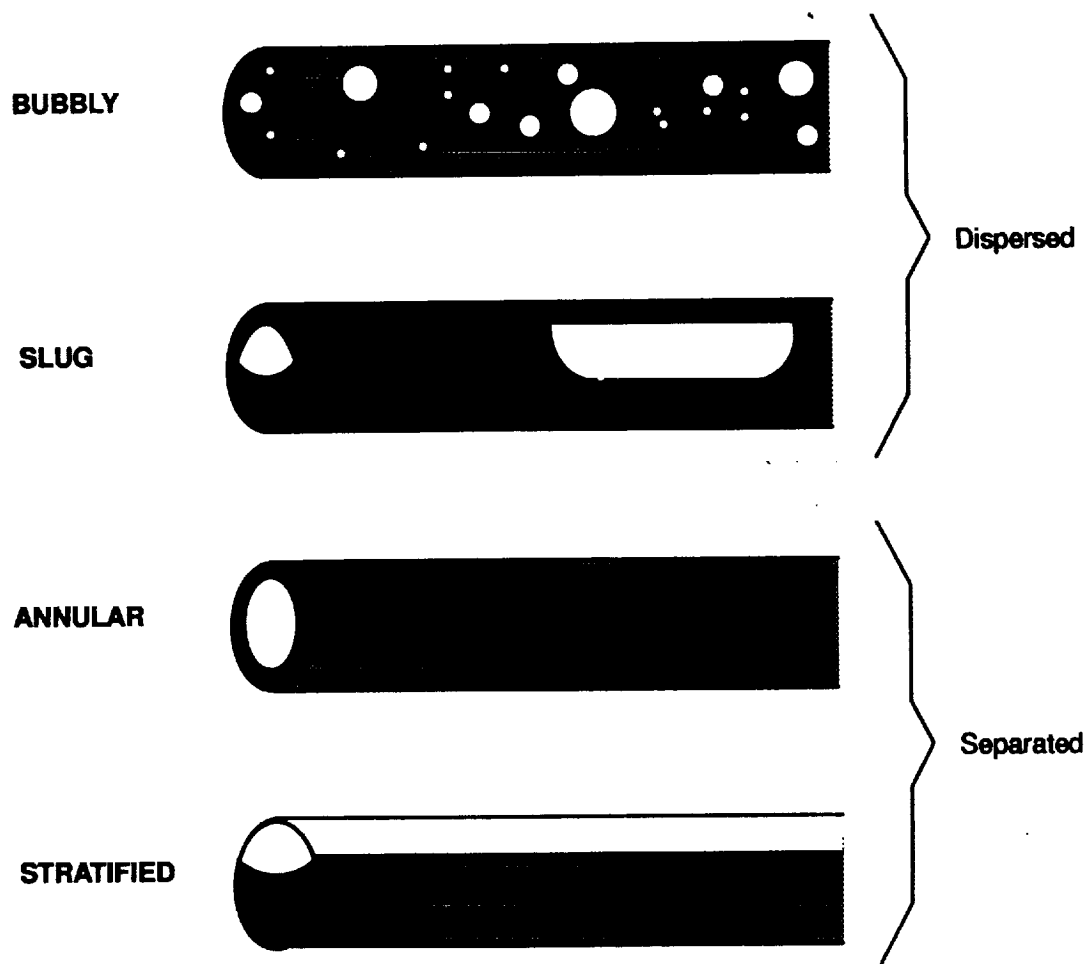


Figure 1 — Simplified Two-Phase Flow Regimes

The annular regime may result if the void fraction continues to grow (above about 0.76), or if the liquid flows downhill, or if there is high enough vapor flux to sustain the uphill flow of liquid. This regime is characterized by a continuous vapor core surrounded and 'lubricated' by a continuous liquid annulus. In most two-phase systems, annular is by far the most common regime. When the regime is determined to be annular, the Lockhart-Martinelli correlation is used.

The stratified regime, characterized by liquid pooling in the bottom of the tube, results if either the vapor mass flux or the liquid fraction is low enough, or the gravity high enough (and the flow is not vertically upward). The stratified regime cannot exist in microgravity. The methods used to predict pressure gradient involve predicting the height of liquid and the fractions of each phase in contact with the wall, assuming a circular cross section (per the method of Taitel and Dukler). Unfortunately, this model is highly sensitive to void fraction, and because the stratified regime typically exhibits the greatest degree of slip, the error in a homoge-

neous approximation to void fraction can be significant. In other words, pressure drops in the stratified regime are suspect if the default homogeneous options are used. Typically, a homogeneous assumption results in overestimation of pressure drop for stratified flow, whereas if slip flow is modeled as described next, the predicted pressure drop is usually lower than that of all other regimes for the same flow quality.

Slip Flow Modeling Options—By default, a homogeneous assumption is applied in all flow passages, meaning that the vapor velocities and the liquid velocities are assumed equal: there is zero relative velocity or *slip*. With the homogeneous approximation, two-phase flow is modeled as the flow of a mixture of both phases—one momentum equation describes the entire duct segment. This assumption is usually adequate and is both simple to implement and fast to execute. Because of this assumption, there is no difference between thermodynamic quality and flow quality. Thermodynamic quality is the fraction of vapor *mass* within a segment divided by the total *mass* in that segment. Flow quality is the ratio of vapor *mass flowrate* through a segment divided by the total *mass flowrate* through that segment.

In reality, vapor usually moves faster than liquid, and sometimes even in opposite directions. A slip flow formulation takes this into account, using one momentum equation per phase. Slip flow options may be applied to any FLUINT duct segment; the homogeneous approximation is retained for pumps, valves, capillary devices, etc.

Unlike homogeneous flow, with slip flow the thermodynamic quality is no longer the same as the flow quality. Conservation of mass dictates that flow quality must be the same (eventually) whether a homogeneous or slip flow formulation is used. However, the thermodynamic quality is no longer constrained by the homogeneous assumption: it becomes the new degree of freedom necessary to accommodate a new momentum equation. In other words, the thermodynamic quality and its manifestations, such as density and void fraction, will vary as needed to balance the flow forces. Because vapor generally travels faster than liquid, *the predicted void fraction will be smaller with slip flow than with homogeneous flow at the same flow quality*. In other words, more liquid will reside in the line, and the thermodynamic quality will be smaller than the flow quality as depicted in Figure 2 for the stratified regime.

Because most pressure drop and heat transfer correlations are based on flow quality, slip flow and homogeneous formulations predict almost the same steady state as long as flow is cocurrent; the local homogeneous assumption does not affect the overall pressure drop and heat transfer rates. The major difference is the proportion of liquid and vapor in lines. For example, in annular flow a slip formulation predicts typically three to four times as much liquid will reside in a pipe compared to a homogeneous prediction. Of course, this amount is small to begin with, and so quoting a factor of three to four might be misleading.

In transients, the differences can be more dramatic, especially for separated flow regimes where vapor can shift quickly and liquid lags behind. As a specific example, a SINDA/FLUINT model was developed to predict the time it takes to clear a small tube of liquid by

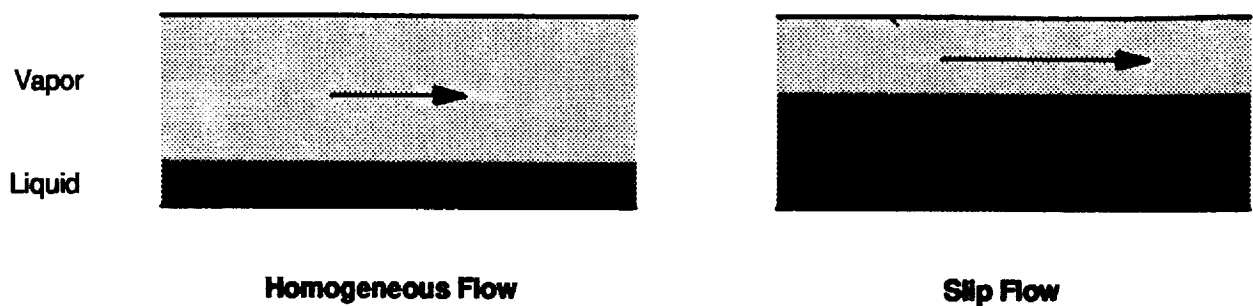


Figure 2 — Homogeneous versus Slip Flow in Stratified Regime at Same Flow Quality

heating it, noting that much more liquid is displaced by generated vapor than is actually evaporated. The default homogeneous assumption resulted in a prediction of 8 seconds to clear the line, whereas allowing slip flow in the same model nearly doubled the duration of the liquid purge event. Since annular flow was quickly established, slip flow allowed the vapor to escape the tube without displacing as much liquid in the process.

This extra degree of modeling power does not come without its price. In addition to greater solution expense, a new layer of uncertainties is revealed. New parameters must be estimated, including (1) the frictional drag between phases, (2) the degree of sharing of inertia, also called added mass and virtual mass, (3) the apportionment of wall friction to each phase, and (4) the momentum transfer associated with phase change. By default, FLUINT will estimate such factors automatically, which requires knowing the flow regime. Hence, flow regime mapping options are defaulted when specifying slip flow. Alternately, like almost all other SINDA/FLUINT options, knowledgeable users can calculate their own coefficients.

Other Improvements—A wide variety of improvements have been implemented to help speed up models utilizing time-dependent fluid elements (called *tanks* and *tubes* in FLUINT) where two-phase flow exist. In general, integrations are smoother, more accurate, and can take larger time steps. Various other improvements have been made in time step predictions, reduced numbers of properties calls, etc., that resulted in speed improvements averaging about 25%.

Also, new simulation options were added to help the user model the mixing of perfect gases, stationary noncondensable gas bubbles, and bellows accumulators. This last option has been applied to other situations requiring two control volumes to share the same physical boundary without exchanging mass.

Because Version 2.5 is to be completed only nine months after 2.4, it was decided that Version 2.4 would be distributed only to NASA centers and Space Station Freedom contractors, and that the next release to COSMIC, who serves a much wider audience, would be delayed until Version 2.5 was completed and tested.

Version 2.5 Enhancements

Three independent improvements have been made in the test version of SINDA/FLUINT Version 2.5, which is scheduled to be officially delivered to NASA JSC in September 1992.

Fast Tabular Fluid Descriptions—It has long been known that one of the significant cost drivers in the solution of fluid is the detail and range of fluid property descriptions. Speed increases can be gained by restricting the description (e.g., providing a liquid-only description) or by simplifying it (e.g., pseudo-perfect vapor equation of state). Also relevant is the fact that ammonia is the most common fluid for two-phase spacecraft thermal management systems. Thus, a new description of ammonia was created that didn't compromise accuracy over the range of temperatures of interest to spacecraft systems, yet runs twice as fast as the built-in ammonia description. This new description uses tabular look-ups, whereas other descriptions describe properties functionally. Once such methods were developed for ammonia, other analogous descriptions were quickly generated for other fluids including hydrogen, nitrogen, oxygen, argon, and ethane.

Single-Phase Heat Transfer with Coarse Discretization—FLUINT slightly underestimates heat transfer for coarsely discretized single-phase lines. This results from assuming an average wall state and an average fluid state over each segment. While such treatment is consistent with the rest of the finite difference (lumped parameter) approximation, which demands nodalization adequate to resolve gradients, it often conflicts with the way many engineers treat a single-phase heat transfer problem: as a constant wall temperature over a segment that has distinct inlet and outlet states. As a result, new heat transfer options have been added to allow such models. For single-phase flows, the predictions are equivalent to a log mean temperature difference (LMTD) solution.

Figure 3 shows how the new methods improve results and/or enable smaller models while yielding the same answers. Comparisons with closed-form solutions are made for this transient thermal/hydraulic analysis of a water pipe with varying inlet temperature and constant wall temperature (Reference 2). To obtain results that are indistinguishable from the closed-form solution, only five control volumes are needed with the LMTD methods compared with twenty for the default downstream-weighted method. Still, the results using traditional methods are good even with only five control volumes. Furthermore, in models of real systems, where gradients in wall temperature or other properties dominate, the differences are usually negligible.

Speeds of Sound and Choking Detection—The user's ability to detect sonic limits was enhanced by providing program options that detect choking in all or portions of a model. The liquid phase remains incompressible, although compressibilities and compressed liquid densities may be calculated and used in concurrent logic, perhaps to calculate effective compliances of control volume walls, or to measure the appropriateness of an incompressible assumption. Other by-products include two-phase speed-of-sound routines.

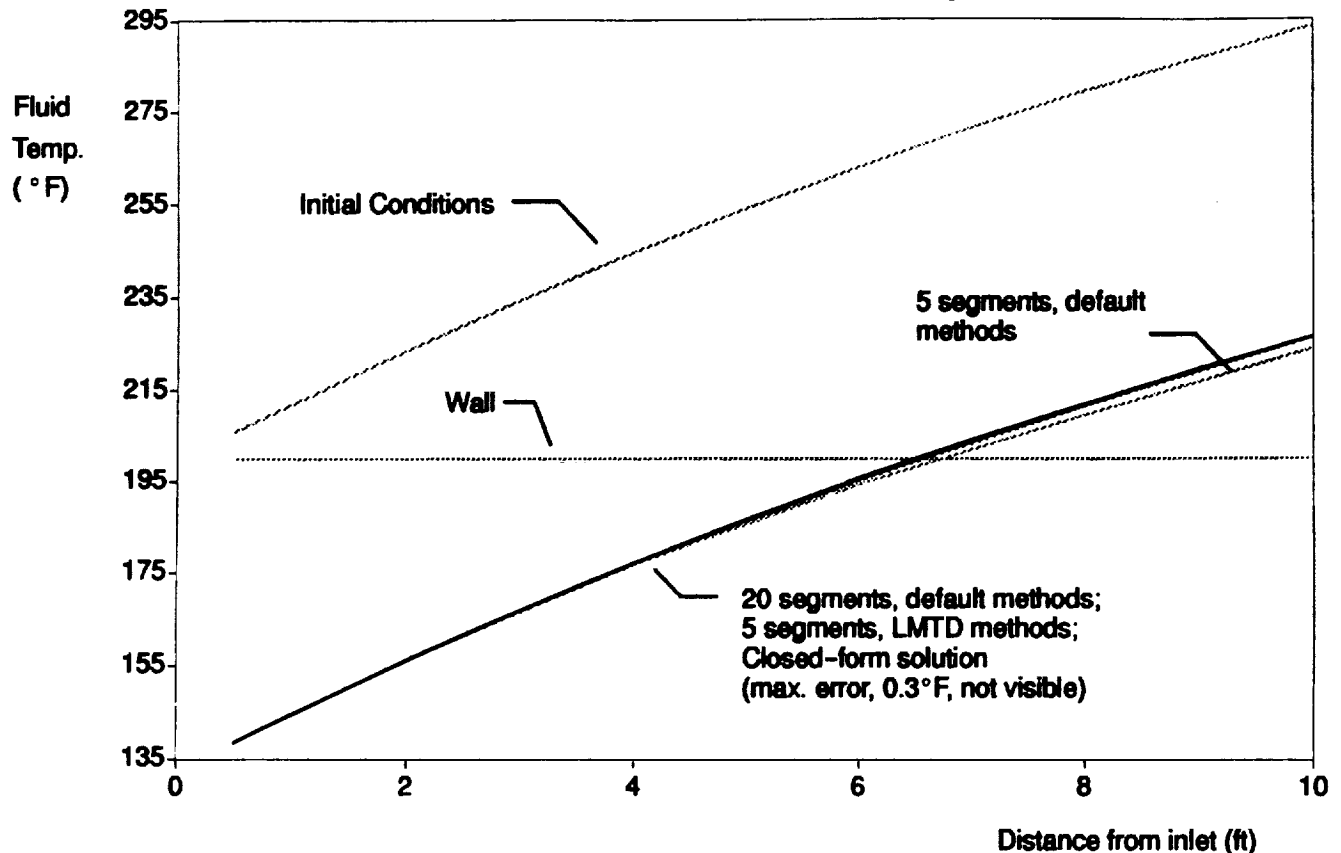


Figure 3 — Comparison of FLUINT Predictions with Closed-form Solution

Other Improvements—Several relatively minor improvements were made in addition to the major thrusts of Version 2.5. These include (1) the addition of K-factors (head loss factors) to duct models, eliminating the need for separate elements to include entrance, bend, and exit losses; (2) the option of using a crash file in addition to normal restart and parametric run options, saving a snapshot of the simulation as often as desired without running out of disk space, (3) reduced memory requirements (matrix inversion work space) for large models, and (4) various internal improvements in time step predictions and slip flow options.

Future Enhancements

Improvements Planned for Version 2.6—FLUINT uses a first-order implicit time step integration that is performed in parallel with whatever method is used to integrate the thermal networks. Heat rates between thermal and fluid models are held constant to conserve energy. If all property domains and derivatives, friction coefficients, heat rates, etc. truly remain constant over the time interval, then the solution is fully implicit and an arbitrarily large time step can be taken. Since these parameters in fact often vary, a best estimate is made of the time step that can be made without excessive changes in such parameters.

Extensive logic is employed to estimate this time step and to check predicted changes against the previous step. While this feedback method successfully avoids time steps that are too small (from the mathematical standpoint if not from the user's standpoint), the only way to be absolutely sure that this estimated time step is not too big is to proceed to integrate the equations and solve for the next network state. If unforeseen changes in operating regimes, boundary conditions, or other parameters are excessive, then at best excessive error will have been generated. At worst, the solution will fail or find a spurious answer such as negative masses in control volumes, or excursions beyond fluid property limits.

In FLUINT, the selected strategy is to spend about 10% to 20% of the cost per solution to make a good and somewhat conservative estimate of the time step. The program is unable to back up and try again if the time step is too big, which fortunately rarely happens. This strategy avoids speed and memory penalties associated with the ability to store and retrieve previous states as well as the problem of trying to measure the generated error and then decide what error is acceptable. A strategy taken in other codes is to take a user-input time step, and then solve iteratively (typically on the order of ten iterations per time step, each about the cost of one FLUINT time step) for the final state. Instead of predicting time steps, the challenge becomes how to converge efficiently on a perhaps elusive final state.

The main thrust of Version 2.6 will be to investigate methods for detecting excessive time steps and correcting them, either by backing up and reducing time steps or by iteratively correcting the solution.

Potential Areas for Future Expansions—Several areas of potential growth have been identified for which no firm development plans exist. These include: (1) full range fluid descriptions with compressible liquid phases that avoid the current discontinuity between saturated liquid and supercritical fluid when the thermodynamic path does not pass through the dome; (2) optionally avoiding the assumption of thermal equilibrium between phases inside of ducts (some limited nonequilibrium capabilities already exist); (3) nonreacting mixtures of substances, especially noncondensable gas phase, air/water systems; (4) higher fidelity capillary models including pore size distributions, wetting hysteresis, partial deprime and liquid recession in the wick; and (5) thermal matrix inversion methods as alternatives to the current iterative closure methods.

SINAPS: SINDA Application Programming System

SINDA/FLUINT, like its predecessors, frees the user from the constraints of real geometry: the model may be limited to a certain volume of material (akin to finite element modeling, or FEM), or it can incorporate a complete vehicle (unlike a finite element approach). The price for this flexibility has been the lack of graphical input and associated postprocessing power, which would help not only in model validation and maintenance, but also in visualization and reporting of results.

Translations to and from solid modeling programs and FEM codes have represented a partial solution for some component design analyses. No analogous capability was present for system-level analyses, or for problems that are intractable with a finite element approach but are amenable to a lumped parameter approach. While postprocessing programs exist to generate X-Y plots of SINDA/FLUINT results, analysts normally communicate with the program via ASCII files. As models grow, the potential for modeling errors or misinterpreted results also grows. (Anecdotally, one small model—a standard sample problem that has been reviewed by many analysts—was found to contain a slight error when rebuilt using SINAPS.)

Nevertheless, hand drawn schematics of SINDA/FLUINT networks are often used to document models in reports and facsimile transmissions. If analysts are able to communicate with each other via such 'artificial' geometry, then it was reasoned they should be able to communicate with the program via similar 2D sketches. After all, similar computer aided engineering packages exist in the electrical design community. Thus, in 1990, NASA JSC initiated an effort to provide a modern graphical pre- and postprocessor for SINDA/FLUINT.

SINAPS is an advanced new companion program to SINDA/FLUINT that enables users to graphically sketch their models using a mouse- and menu-driven user interface. Forms and editing windows exist to satisfy other nongraphic SINDA/FLUINT input requirements. SINAPS then produces complete SINDA/FLUINT ASCII input files, and imports binary output files that were perhaps produced on other machines. This enables *graphical display of predictions on the same schematic used to create inputs*. In addition to pop-up X-Y, polar, and bar plots, features such as "color by flowrate," "thicken by conductance," and "shade by temperature" are supported. Figures 4 and 5 present two sample SINAPS screen images (in black and white for reproduction) that depict some of the features available.

SINAPS is intended to become a complete, modern front-end to SINDA/FLUINT, eliminating the need to communicate via ASCII input and output files. In fact, it contains many powerful features that are unavailable in the basic SINDA/FLUINT system, such as algebraic inputs, shared models, customized components, etc. To assist current SINDA/FLUINT users in the transition to SINAPS, it will accept existing ASCII input files, and will work interactively with the user to produce a graphical depiction of that model.

SINAPS is transportable. It was developed simultaneously on a Macintosh II and a SUN SPARCstation, and can be rehosted on most other workstations, operating systems, and windowing systems. Perhaps more importantly, a SINDA/FLUINT model (and its graphical depiction) built using SINAPS can be easily moved from one type of machine to another, allowing analysts to build models on whatever machines are available, even if that availability changes from day to day. Combining this feature with the fact that SINAPS and SINDA/FLUINT need not reside on the same machine gives the analyst tremendous flexibility.

SINAPS will be available in the fall of 1992, and will correspond to SINDA/FLUINT Version 2.5.

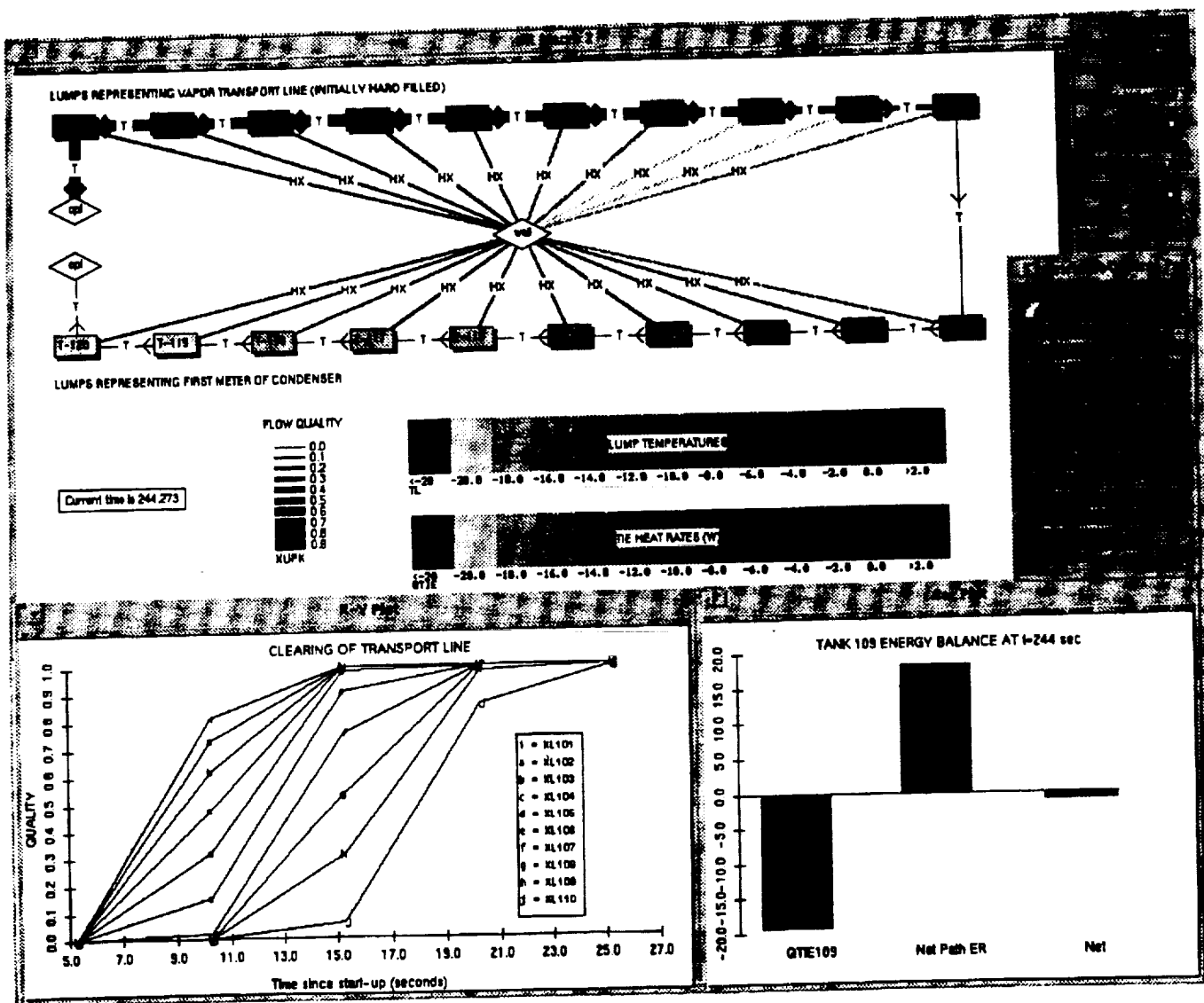


Figure 4 — Sample SINAPS Screen: Capillary Pump Loop Start-up Transient

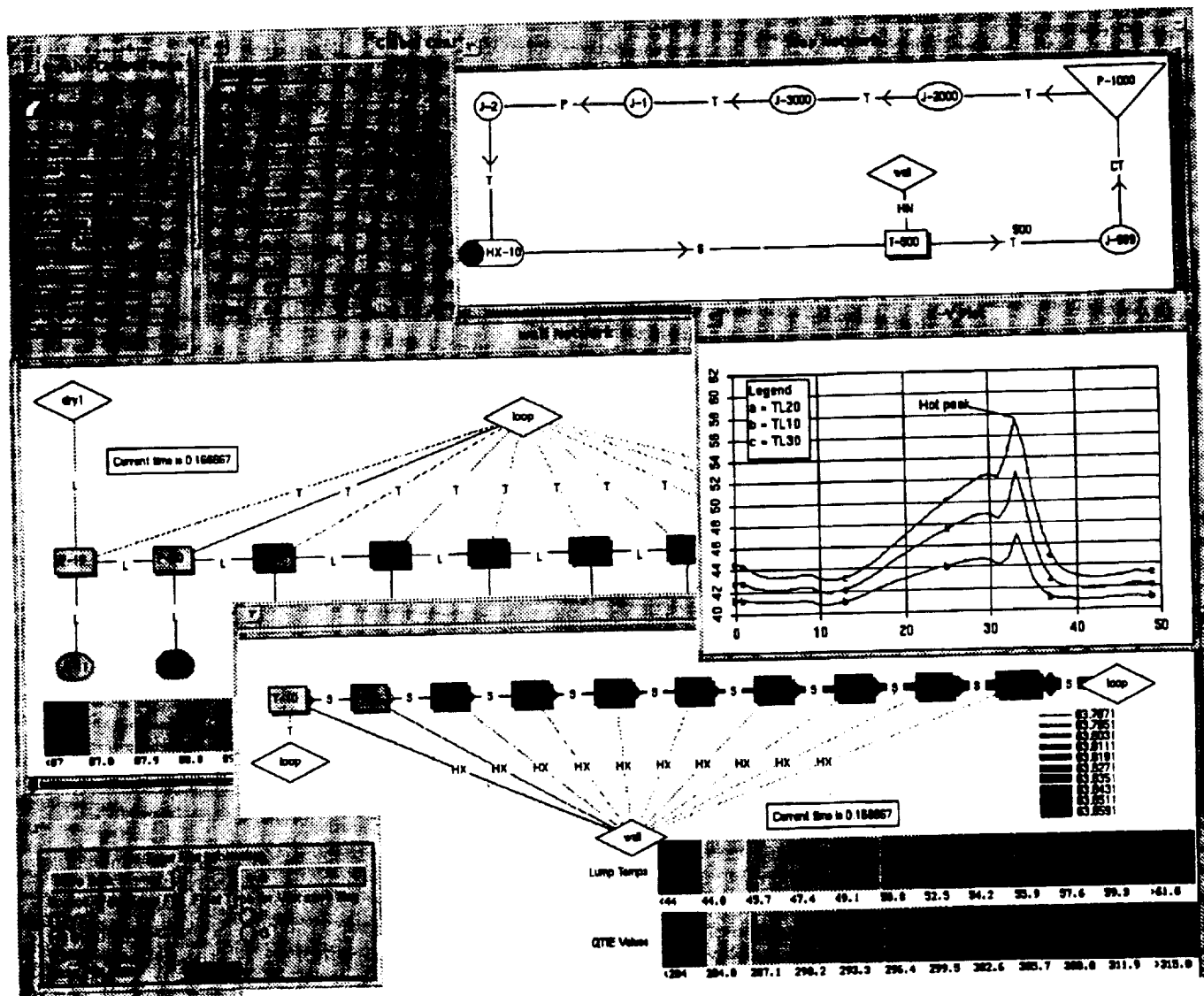


Figure 5 — Sample SINAPS Screen: Pump Instabilities with Temperature Control Valve in Series

Acknowledgements

Neither SINDA/FLUINT nor SINAPS would exist were it not for the continuing support of the NASA Johnson Space Center. These codes were produced by Martin Marietta under contracts NAS9-17053, NAS9-17448, and NAS9-18411.

References

1. Cullimore, B.A.; and Lin, C.H.: FLUINT: Generalized Fluid System Analysis with SINDA '85. AIAA-87-1466; AIAA 22nd Thermophysics Conference, June 1987.
2. Cullimore, B.A.: Applications of a Generalized Thermal/Hydraulic Simulation Tool. AIAA-89-1754; AIAA 24th Thermophysics Conference, June 1989.
3. Cullimore, B.A.; Ring, S.G.; Goble, R.G.; and Jensen, C.L.; SINDA '85/FLUINT Systems Improved Numerical Differencing Analyzer and Fluid Integrator Version 2.3, User's Manual; MCR 90-512, Contract NAS9-17448; March, 1990.
4. Cullimore, B.A.; Ring, S.G.; Goble, R.G.; and Jensen, C.L.; SINDA/FLUINT Systems Improved Numerical Differencing Analyzer and Fluid Integrator Version 2.4, User's Manual; MCR 91-1393, Contract NAS9-18411; December, 1991.
5. Crowley, C.J.; and Izenson, M.G.; Design Manual for Microgravity Two-Phase Flow and Heat Transfer; AL-TR-89-027; October, 1989.

A SINDA THERMAL MODEL USING CAD/CAE TECHNOLOGIES

Jose A. Rodriguez*
Sverdrup Technology, Inc.
Huntsville, AL

Steve Spencer†
Sverdrup Technology, Inc.
Huntsville, AL

SUMMARY

The approach to thermal analysis described by this paper is a technique that incorporates Computer Aided Design (CAD) and Computer Aided Engineering (CAE) to develop a thermal model that has the advantages of Finite Element Methods (FEM) without abandoning the unique advantages of Finite Difference Methods (FDM) in the analysis of thermal systems. The incorporation of existing CAD geometry, the powerful use of a pre and post processor and the ability to do interdisciplinary analysis, will be described.

INTRODUCTION

Since the birth of the Chrysler Improved Numerical Differencing Analyzer (CINDA), a tool widely used in the aerospace industry, many improvements to the code have been made. Lately, the advances have been dramatic, starting with the rewrite of the Systems Improved Numerical Analyzer (SINDA) to what is now known as SINDA '85, to the addition of fluid analysis and the creation of a graphical interface, SINDA Application Programming System (SINAPS). Most of the progress has concentrated around the Finite Differencing methods, with very slow progress in the graphical end of the analysis.

*Engineer, Sverdrup Technology, Inc., MSFC Group.

†Senior Engineer, Sverdrup Technology, Inc., MSFC Group.

Computer aided Engineering (CAE) have revolutionized the analytical world. Most of the advances have taken place in the field of Finite Element Methods (FEM). FEM, an ideal tool for structural analysis, is not well suited for thermal analysis yet, specially, when the problem is radiation dominated. FDM because of its particular characteristics to handle nonlinear systems, has been the method of preference in the analysis of thermal systems. The advantages of CAE created an interest in the thermal analysis discipline that gave way to translators that can convert FEM format to FDM format, thus creating a unique opportunity for the thermal analyst.

This paper concentrates on the methodology of using CAD generated geometry in a CAE environment to develop a thermal model. The format and mathematics used on both, CAD and CAE platforms, is different requiring therefore a translator to share the information. The Initial Graphics Exchange Specification (IGES) version 5.0 translator was selected due to its availability on both platforms.

THERMAL MODEL UPDATE FOR THE AEROASSIST FLIGHT EXPERIMENT (AFE)

The thermal model of the AFE had reached a level of obsolescence and a major update with required. As the design matured, there was a need to update the model to include all design reviews. The structural design had major changes, but it was still in the evolution stage. To update the model, a major undertaking was necessary, but there was still the question of how to keep up with the changes, especially, how to respond to the "what if" questions that were being posed as changes in the design were taking place? There was a choice of doing the update the traditional way and lag behind or investigate the avenue of using the CAD generated geometry to create the model and to incorporate updates of components as they change, without affecting the rest of the model.

A NEW APPROACH IN THERMAL ANALYSIS

The incorporation of CAD generated geometry to create a thermal model in a CAE environment, is an avenue that will complement established practices and it will also allowed the analyst to do the following:

- Use the same geometry generated by the designer

- Eliminate design mis-interpretation
- Avoid dimensional errors
- Maintain model fidelity
- Update design changes only on the affected areas
- Share the model with other disciplines
- Share results with other disciplines
- Promote concurrent engineering

The design of the AFE was done using the Intergraph Graphics Design System (IGDS) and the Intergraph Engineering Modelling System (I/EMS) packages. Use of the CAD data required transferring it to the CAE platform, where the actual modelling would take place and finally to a platform where the conversion to a SINDA model is done. Following are the steps taken to create a thermal model using CAD data.

- Compress the boolean trees of the CAD file
- Using the I/IGES translator, translate the I/EMS file to an IGES file
- Using the CADPAT, IGES-to-PATRAN translator, translate the IGES file to a PATRAN neutral file.
- Prepare the model in PATRAN by defining the nodal network, physical properties and material properties.
- Translate the model (a PATRAN neutral file) to a FEM/SINDA file, using the FEM-to-FDM translator FEM/SINDA.
- Using FEM/SINDA, convert the FEM/SINDA file to a Finite Difference file.

Table 1 shows the geometry entities I/IGES version 5.0 and CADPAT release 4.0, can support.

Table 1. IGES Entities Supported by I/IGES and CADPAT

I/IGES	CADPAT	IGES Entity & Number
Circular arc or circle	Parametric cubic line	Circular arc, 100
Composite curve	Parametric cubic line	Composite curve, 102
B-Spline curve	Parametric cubic line	General conic, 104
Points	Parametric cubic line	Data points, 106
B-Spline curve	Parametric cubic line	Parametric spline curve, 112
B-Spline surface	Parametric cubic patch	Parametric spline surface, 114
Point	Grid	Point, 116

B-Spline surface	Parametric cubic patch	Ruled surface (arc length),118
B-Spline surface	Parametric cubic patch	Surface of revolution, 120
B-Spline surface	Parametric cubic patch	Tabulated cylinder,122
Transformation matrix	Coordinate system	Transformation matrix,124
B-Spline curve	Parametric cubic line	Rational B-Spline curve,126
B-Spline surface	Parametric cubic patch	Rational B-Spline surf.,128
B-Spline surf. boundary	Primitive parametric surface curve	Curve on parametric surface, 142
B-Spline surface	Primitive face	Trimmed parametric surface, 144

The I/EMS model of the AFE, figure 1, is composed of many components, due to its massive size, it was necessary to separate each component in individual files, making the translation process less cumbersome. Once the files were translated, figure 2, and modeled individually as a component, they were merge to form the complete model. Figure 3 depicts a flowchart of the translation process.

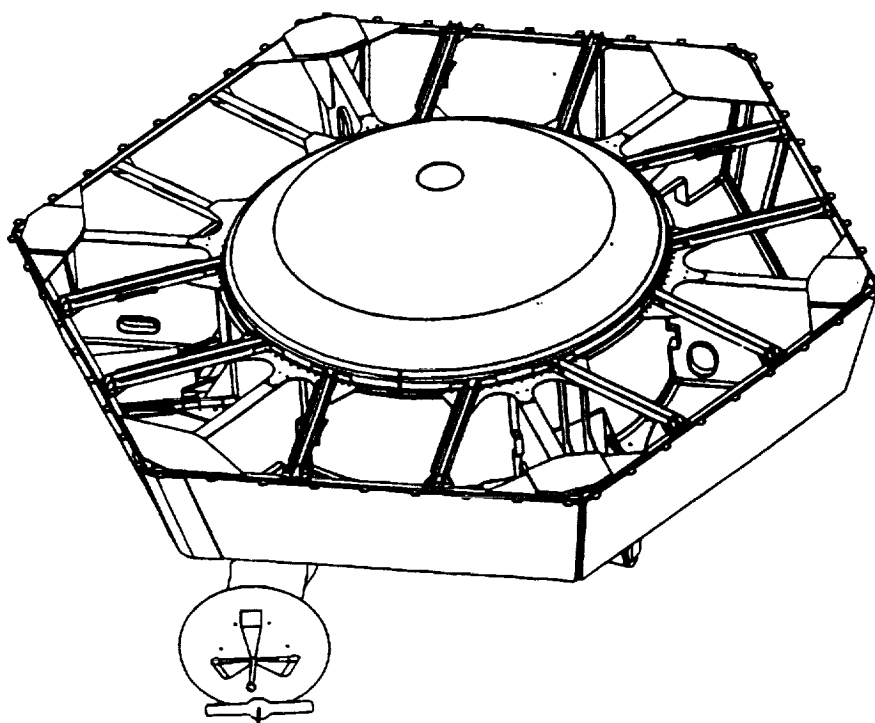


Figure 1. Aeroassist Flight Experiment CAD drawing

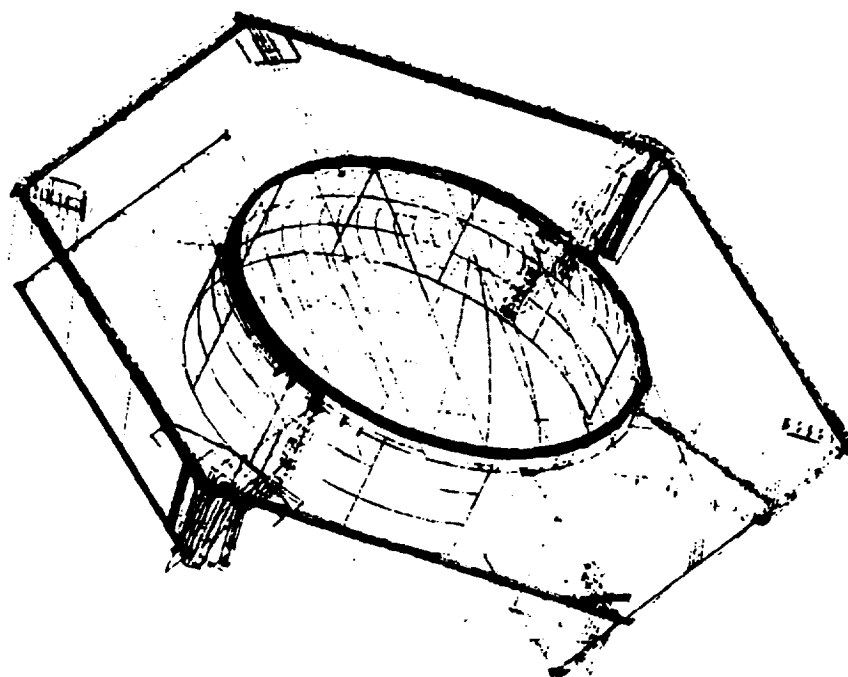


Figure 2. Aeroassist Flight Experiment translation from CAD to CAE, without modifications

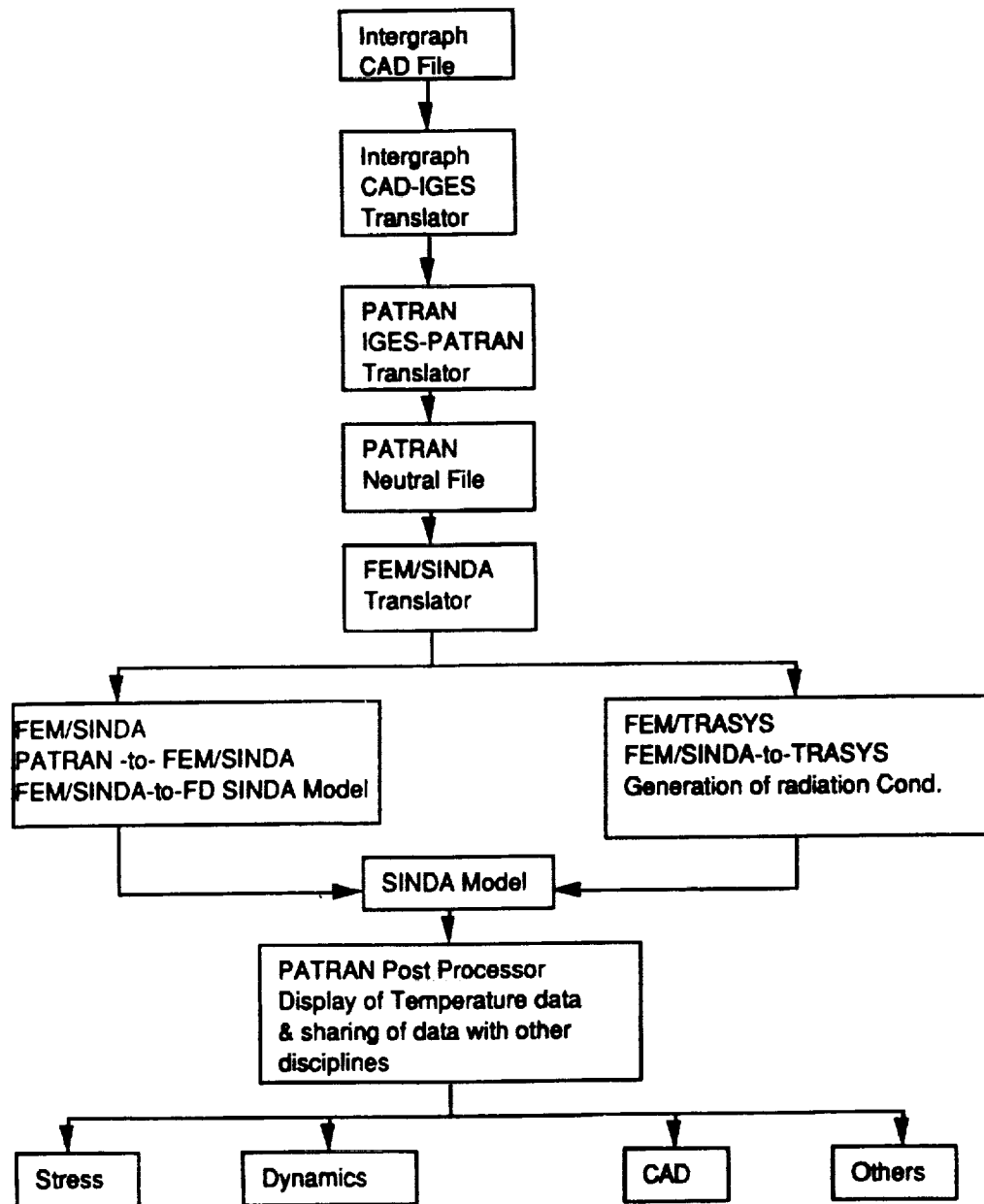


Figure 3. Thermal Analysis CAD/CAE Interdisciplinary Illustration For The AFE Experiment

CONCLUSIONS

The process of using CAD/CAE technologies is not a substitute for the traditional thermal analysis, but merely a complement to the task of analyzing thermal systems. It is a process that can be applied when dealing with large and complicated structures, when hand inputs can take several months. This process goes beyond thermal analysis, this process encourages the members of all disciplines working in a project, to work together, the results of such cooperation are unlimited, with sharing of models and results being just a few.

Due to cancelation of the AFE project, the complete modelling of the AFE using this process was not possible, but the major components were incorporated in the model. Further work is progressing through other projects. Presently this process is being applied to the creation of SINDA and TRASYS models from the same CAD/CAE file, as well as creating geometry from TRASYS models that were written by hand and needed updating. New technologies and cooperation between CAD and CAE vendors will expedite this process.

ACKNOWLEDGEMENTS

The present process investigation would not have been possible without the support of the AFE thermal analysis project manager, Mr. Patrick Hunt of NASA Marshall Space Flight Center, who encouraged the investigation of another avenue for the model update, Dr. W. Randolph Humphries of NASA Marshall Space Flight Center, whose interdisciplinary charter help us investigate the process with different structures and our supervisor, Dr. Alok Majumdar, who kept on encouraging us throughout the investigation.

REFERENCES

1. Initial Graphics Exchange Specification, version 5.0, U. S. Department of Commerce.
2. CADPAT, release 4.0, PDA Engineering.
3. FEM/SINDA User's manual, version 2.0, Martin Marietta Corporation Missile Systems., Orlando, Florida.

A Compressible Boundary Layer Algorithm for Use with SINDA '85

Barbara Sakowski and Douglas Darling
NASA - Lewis Research Center

Allan van de Wall
Case Western Reserve University

NOMENCLATURE

C_p	specific heat at constant pressure
i	index of grid location in the x-direction
j	index of grid location in the y-direction
jj	total number of grid points in the y-direction
P	pressure
r	radius of wall (axisymmetric case)
Re	Reynolds Number (per unit length)
RUB	coefficient in front of derivatives of the advective terms in the x-direction
RVB	coefficient in front of derivatives of the advective terms in the y-direction
T	temperature
u	velocity component in direction of core flow
v	velocity component perpendicular to core flow
x	location in the direction of the core flow
y	location in the direction perpendicular to the core flow
μ_{eff}	total viscosity (molecular and turbulent)
ρ	density
max	core flow value at point with maximum Mach number, used to non-dimensionalize boundary layer equations

INTRODUCTION

At times we need to analyze the thermal behavior of systems that include both conduction and high speed flows. Unfortunately, most high-speed-flow codes have limited conduction capabilities and most conduction codes, such as SINDA, cannot model high speed flows. It would be useful to interface a high-speed-flow solution and SINDA. When interfacing a high-speed-flow solution to SINDA, it may be necessary to include the viscous effects in the energy equations. Boundary layer effects of interest include heat transfer coefficients (including convection and viscous dissipation) and friction coefficients. To meet this need, a fast, uncoupled, compressible, two-dimensional, boundary layer algorithm was developed that can model flows with and without separation. This algorithm was used as a subroutine with SINDA. Given the core flow properties and the wall heat flux from SINDA, the boundary layer algorithm returns a

wall temperature to SINDA. SINDA and the boundary layer algorithm are iterated until they predict the same wall temperature.

BOUNDARY LAYER ALGORITHM

Boundary Layer Equations

The forms of the boundary layer equations used in the finite difference scheme were the compressible, parabolized Navier-Stokes (PNS) equations by Roach, et al [1]. The equations were 2-dimensional, viscous, and were solved for the primitive variables. The y-momentum equation simply reduced to

$$\frac{\partial \bar{P}}{\partial y} = 0 \quad (1)$$

The other equations were as follows:

continuity

$$\frac{\partial(\bar{\rho} \bar{u})}{\partial x} + \frac{\partial(\bar{\rho} \bar{v})}{\partial y} = 0 \quad (2)$$

x momentum

$$\bar{\rho} \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{\rho} \bar{v} \frac{\partial \bar{u}}{\partial y} = \frac{1}{R_o} \frac{\partial}{\partial y} \left(R_o \frac{\partial \bar{u}}{\partial y} \right) - \frac{\partial \bar{P}}{\partial x} \quad (3)$$

energy

$$\bar{\rho} \bar{u} \frac{\partial \bar{T}}{\partial x} + \bar{\rho} \bar{v} \frac{\partial \bar{T}}{\partial y} = \frac{1}{R_o} \frac{\partial}{\partial y} \left(\frac{R_o}{RePr} \frac{\partial \bar{T}}{\partial y} \right) - \frac{1}{Re} \left(\frac{\partial \bar{u}}{\partial y} \right)^2 + \bar{u} \frac{\partial \bar{P}}{\partial x} \quad (4)$$

The variables were defined as:

$$\bar{\rho} = \frac{\rho}{\rho_{\max}} \quad (5)$$

$$\bar{u} = \frac{u}{u_{\max}} \quad (6)$$

$$\bar{P} = \frac{P}{\rho_{\max} u_{\max}^2} \quad (7)$$

$$\bar{T} = \frac{c_p T}{u_{\max}^2} \quad (8)$$

$$Re = \frac{\rho_{\max} u_{\max}}{\mu_{\text{eff}}} \quad (9)$$

For axisymmetric flows, ($k=1$), R_0 was defined as:

$$R_0 = r \pm y \cos \theta \quad (10)$$

where r was the radius of the wall, θ was the angle of the wall, and the plus or minus signs referred to external or internal flows respectively. For rectangular coordinates R_0 was ignored, ($k=0$). The coordinates x and y used in the above equations were the transformed x and y coordinates, that is, the x coordinate followed along the wall surface and the y coordinate was perpendicular to the wall. It should be noted that Re was not actually non-dimensional. Re had units of $(\text{length})^{-1}$. Also, x and y (and R_0) remained dimensional. So, each term in each of the above boundary layer equations had units of $(\text{length})^{-1}$.

Solution Algorithm

The momentum, continuity and energy equations were differenced as described by Kwon et al.[2] For any scalar quantity ϕ :

$$\left(\frac{\partial \phi}{\partial x} \right)_{ij} = \frac{\phi_{i,j} - \phi_{i-1,j}}{x_i - x_{i-1}} \quad (11)$$

$$\left(\frac{\partial \phi}{\partial y} \right)_{ij} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{y_{j+1} - y_{j-1}} \quad (12)$$

$$\left(\frac{\partial}{\partial y} \frac{1}{Re} \frac{\partial \phi}{\partial y} \right)_{ij} = \frac{\frac{1}{Re_{i,j+\frac{1}{2}}} \frac{\phi_{i,j+1} - \phi_{i,j}}{y_{j+1} - y_j} - \frac{1}{Re_{i,j-\frac{1}{2}}} \frac{\phi_{i,j} - \phi_{i,j-1}}{y_j - y_{j-1}}}{\frac{1}{2}(y_{j+1} - y_{j-1})} \quad (13)$$

Due to the parabolic nature of the boundary layer equations, the governing

equations were solved by marching from station to station in the direction of the core flow. Solutions at a given station were obtained by solving the boundary layer equations sequentially. First, the momentum equation was solved for the velocity component in the core flow (streamwise) direction (u). Second, continuity was solved for the velocity component in the y direction (v). Then, the energy equation was solved for the temperature (T). If the station had not converged, the momentum, continuity, and energy equations were solved again for the velocity components and temperature, using the flow properties from the previous iteration. Convergence at a given station was obtained when the streamwise velocity components at all grid locations at that station converged. After the calculation at the station converged, the algorithm marched to the next streamwise station to solve for the boundary layer properties. This streamwise marching continued throughout the entire domain.

Grid spacing perpendicular to the wall (y-direction) was based on an exponential function. The grid spacing was fine near the wall to better resolve the gradients at the wall. The grid spacing was coarse away from the wall where fine resolution of the gradients normal to the core stream were not necessary.

The derivative of the velocity at the wall (used to calculate wall shear stress) was determined using a second order approximation. Values at $j=1$ (wall), $j=2$, and $j=3$ were used.

$$\left(\frac{\partial u}{\partial y}\right)_{i,1} = \frac{y_{i,3}^2(u_{i,2} - u_{i,1}) - y_{i,2}^2(u_{i,3} - u_{i,1})}{y_{i,2}y_{i,3}(y_{i,3} - y_{i,2})} \quad (14)$$

The above algorithm worked well for flows with weak viscous/inviscid interaction, since the downstream influence could be neglected. However, when the flows were strongly interacting, such as those with strongly adverse pressure gradients or separation, downstream conditions had to be considered. To account for downstream influences the pressure gradient was differenced as a weighted average of forward and backward differences. This technique was based on the method of Davis and Barnett[3].

Specifically the pressure gradient was differenced as

$$\frac{\partial \bar{P}}{\partial x} = (\epsilon) \left(\frac{\bar{P}_{i,j} - \bar{P}_{i-1,j}}{x_i - x_{i-1}} \right) + (1-\epsilon) \left(\frac{P_{i+1,j} - P_{i,j}}{x_{i+1} - x_i} \right) \quad (15)$$

where ϵ is a weighting parameter of the forward and backward differencing. This term was required to remove the ellipticity in the PNS equations in strongly interacting flows. The quantity ϵ determined what fraction of the forward difference of the pressure

gradient can be included so that the equations remain non-elliptic. If the flow was supersonic at a given j location, then only backward differencing was used for the pressure gradient ($\epsilon = 1$). If the flow was subsonic then at a given j location then the following expression for ϵ was used,

$$\epsilon = \frac{\gamma M^2}{1 + (\gamma - 1)M^2} \quad (16)$$

Boundary Layer Separation

Flaring of the advective term parallel to the core flow was used when the flow was separated. This was handled by taking the absolute value of the coefficient in front of the derivative of the advective term (RUB) and multiplying it by .1 to make it smaller. Again this operation was only performed when the flow was separated.

Boundary Layer Turbulence Model

A modified Baldwin-Lomax model was used to account for turbulence. This model was a zero equation, eddy viscosity model. This model was faster than other turbulence models, such as the $k-\epsilon$ two equation model. A modified model of Visbal and Knight was used to handle the effects of separation. However, the model of Visbal and Knight required further modification, the most important of which was the modification of the Baldwin-Lomax parameter C_φ . C_φ needed to be a function of both core flow Mach number and core flow pressure gradient. This modification was extremely important in matching numerical results and experimental data. A detailed description of the turbulence model used is given in Sakowski, et. al [4].

Boundary Layer/Core Flow Interface

The boundary layer algorithm serves as a link between the conduction program (SINDA) and an inviscid core flow program. The interface with SINDA will be discussed in a later section. In this section we will look at the interface of the boundary layer algorithm with a core flow program.

When the boundary layer algorithm was interfaced with a core flow algorithm, it was necessary that the boundary layer properties smoothly approached the core flow values. That is, when the derivatives perpendicular to the wall were zero, the core flow values had to be a solution to the boundary layer equations. The complicated part was matching the boundary layer and core flow, since the boundary layer and core flow algorithms were probably not differenced in the same way. If the differential equations were solved exactly there would not be problem, but they were not solved exactly. What the core flow algorithm predicted as a solution, was not exactly what the boundary layer algorithm predicted as a solution as the y -derivatives went to zero (far from the wall).

The difference was usually fairly small (2% or so), but this small difference could have a big effect on the integral performed to calculate the displacement thickness. An adjustment in the way the pressure gradient term was calculated in the boundary layer algorithm forced the boundary layer properties to smoothly approach the core values. Without this adjustment, the displacement thickness, predicted by the boundary layer program had large errors. The adjustment of the pressure gradient was performed by solving the finite differenced momentum equations for dP/dx when all y -derivatives were zero.

$$\frac{\partial \bar{P}}{\partial x} = -\bar{\rho}_e \bar{u}_e \frac{\partial u_e}{\partial x} \quad (17)$$

The corrected value of the pressure derivatives was calculated with the edge values from the core flow algorithm using the same differencing scheme used in the boundary layer algorithm. In this way the boundary layer algorithm approached the core flow values as the y -derivatives approached zero.

Another consideration for interfacing a core flow algorithm and the boundary layer code was stability. At times the boundary layer algorithm has a stability problem. This problem tended to initiate near the edge of the boundary layer. From one iteration to the next the values near the core flow sometimes fluctuated between less than and greater than the core flow value. Sometimes these fluctuations died out and the program converged. However, other times the oscillations grew, causing the calculations to diverge. To solve this problem flaring was used. RVB was part of the advective terms in the y -direction. RVB was the coefficient in front of the $\partial u / \partial y$ term in the finite differenced x -momentum equation, and the $\partial T / \partial y$ term in the finite differenced energy equation. These were the advective terms in the y -direction. Without flaring RVB was simply ρv . With flaring RVB was changed as follows:

$$RVB = K_1 | \bar{\rho} \bar{v} | \quad (18)$$

K_1 's for momentum equation

$$\text{if } u < u_e \quad \text{if } u > u_e$$

$$\text{if } \frac{\partial u}{\partial y} > 0 \quad K_1 = -\frac{u}{u_e} \quad K_1 = \left(1 - \frac{u}{u_e} \right)$$

$$\text{if } \frac{\partial u}{\partial y} < 0 \quad K_1 = \left(1 - \frac{u}{u_e} \right) \quad K_1 = -\frac{u}{u_e}$$

K_1 's for energy equation

if $T < T_e$ if $T > T_e$

$$\text{if } \frac{\partial T}{\partial y} > 0 \quad K_1 = -\frac{T}{T_e} \quad K_1 = \left(1 - \frac{T}{T_e}\right)$$

$$\text{if } \frac{\partial T}{\partial y} < 0 \quad K_1 = \left(1 - \frac{T}{T_e}\right) \quad K_1 = -\frac{T}{T_e}$$

The motivation for the above flaring was to make the core flow value a numerically stable solution in the boundary layer algorithm far from the wall. This flaring was found to be very important to help the stability of the algorithm, particularly when there was an adverse pressure gradient, separation, bleeds, or bypasses. For more detail on interfacing the boundary layer algorithm of Roach, et al. with a core flow, refer to Darling, et al. [5].

CORE FLOW INPUT TO BOUNDARY LAYER ALGORITHM

It was mentioned previously that the boundary layer algorithm required as an input, a core flow. The inviscid core flow variables needed by the boundary layer algorithm were the Mach number, temperature, and pressure. For flows where the interaction between the core flow and boundary layer were negligible, the interface was very simple and easily implemented. All that was required was an input file by the name LBL.DAT. This file contains seven namelists described as follows:

NAMELIST/GEOM/X:	Grid point locations that run parallel to the centerline of the wall
NAMELIST/AREAX/AREA:	Surface area of the wall that corresponds to a respective X grid point location
NAMELIST/TEMP/T1:	Core flow temperature that corresponds to a respective X grid point location
NAMELIST/PRES/P1:	Core flow that corresponds to a respective X grid point location

NAMELIST/MACH/AM: Core flow Mach number that corresponds to a respective X grid point location

NAMELIST/RADIUS/RADY: Radius of the wall measured from the wall centerline and perpendicular to its respective X grid point location

NAMELIST/BL/

JJJ: Number of grid points perpendicular to the wall

DUMX: Convergence criteria on the u velocity component of the x momentum equation

II: Number of grid points in the streamwise direction

TURB: Flag to signal use of the Baldwin-Lomax turbulence model:
TURB=.TRUE. - turbulence on
TURB=.FALSE. - turbulence off

ENGU: Flag to signal use of units:

ENGU	T1	P1	X	AREA
ENG UNITS - .TRUE.	*R	lbf/ft²	ft	ft²
SI UNITS - .FALSE.	*K	N/m²	m	m²

AXI: Flag to signal axisymmetric flow:
AXI=.TRUE. - axisymmetric flow
AXI=.FALSE. - 2-D flow

EXT: Flag to signal external flow:
EXT=.TRUE. - external flow
EXT=.FALSE. - internal flow

MYES: Flag to print to:
Mach.out: Prints non-dimensional velocity, temperature, density, and pressure profiles for each x grid location and for the first 10 SINDA iterations - after 10 SINDA iterations, the profiles are printed every 20 SINDA iterations

Each x grid location also gives the following parameters:
IT: number of boundary layer iterations
M: core flow mach number
Cf: skin friction coefficient
Ybl: distance of the last y grid point from the wall

C2.OUT: Prints the following boundary layer parameters:
IT: number of boundary layer iterations
Th: momentum thickness
Cf: skin friction coefficient
Disp: displacement thickness
for the first 10 SINDA iterations - after 10 SINDA iterations, the profiles are printed every 20 SINDA iterations

A sample LBL.dat file is provided in the APPENDIX.

For flows where there was a strong interaction between the core flow and boundary layer, the simple input data file of constant inviscid core flow values would change complexion. The file would become a core flow algorithm which continuously updates the core flow variables to account for the boundary layer interaction, such as flows with shock waves. Such algorithms are not discussed in this paper.

SINDA/BOUNDARY LAYER INTERFACE

The nodes in SINDA that represent the wall surface nodes **MUST** be declared as boundary nodes. This is done to obtain heat rates on the nodes that can be sent to the boundary layer algorithm. For a steady state solution the heat rate on the surface node will be zero if the nodes are defined as arithmetic or diffusion nodes. This is because SINDA effectively "sees" an insulated surface. In actuality the surface is not insulated, because of the presence of the boundary layer flowing over it. For each SINDA iteration, the heat rates from the SINDA nodes are passed to the boundary layer where new wall temperatures are determined. The boundary layer algorithm will continue iterating until it has a converged solution. After the boundary layer algorithm has converged, the boundary layer wall temperatures become the new SINDA boundary node temperatures. Thus the SINDA boundary node temperatures are updated every SINDA

iteration. The program iterates between SINDA and the boundary layer algorithm until the boundary layer wall temperatures match the SINDA boundary node temperatures.

SINDA INPUT TO BOUNDARY LAYER ALGORITHM

In **HEADER CARRAY DATA**, reserve the following variables:

1=LBL.DAT
2=MACH.OUT
3=C2.OUT

Also, the common SINDA/FORTRAN variable **BTEST** should not be used. In **HEADER VARIABLES 1**, make the call to the subroutine **INTERFACE** as follows:

CALL INTERFACE(A,B,C,D,E,ABSZRO,BTEST,UCA1,UCA2,UCA3)

where **A** = Submodel name in quotes where the surface boundary nodes are located, and no more than 8 characters long.

B = SINDA node number of the surface boundary node located at the last x grid location of the boundary layer (integer).

C = SINDA node number of the surface boundary node located at the first x grid location of the boundary layer (integer).

D = Number by which the SINDA boundary nodes are incremented (integer).

E = Units used by SINDA. 'ENG' for English units. 'SI' for SI units.

The remaining arguments should be left as they are. **ABSZRO** is the SINDA variable for absolute temperature defined in the **HEADER OPTIONS DATA BLOCK**. **BTEST** is a counter for the number of SINDA iterations. **UCA1**, **UCA2**, and **UCA3** are the SINDA variables for the input and output file names defined in the **HEADER CARRAY DATA BLOCK**.

CONCLUSIONS

A fast steady, compressible, turbulent boundary layer algorithm that can be used to model separated flows has been written as a subroutine for SINDA. Results from the boundary layer algorithm compared well with experimental pressure distributions when the boundary layer was interactive with the core flow, Darling, et al. [5] and Roach, et al

[1]. In addition, the friction coefficients and momentum thicknesses predicted by the boundary layer code compared well with experimental data, Roach, et al. [1]. The Baldwin-Lomax turbulence model was used following the modifications of Sakowski, et al [4], also matched experimental data fairly well. Currently comparisons are being made with experimental data validate the heat transfer predictions of the boundary layer algorithm. The boundary layer algorithm was found to converge quickly with SINDA. A simple SINDA model with 25 x-grid locations was tested, see the APPENDIX for the SINDA input. The model converged in 95 SINDA iterations.

REFERENCES

1. R.L. Roach, C. Nelson, B.A. Sakowski, D.D. Darling, A.G. van de Wall, "A Fast, Uncoupled, Compressible, Two-Dimensional, Unsteady Boundary Layer Algorithm with Separation for Engine Inlets," AIAA 92-3082, 1992.
2. O.K. Kwon, R.H. Pletcher, R.A. Delaney, "Solution Procedure for Unsteady Two-Dimensional Boundary Layers," ASME Journal of Fluids Engineering, Vol. 110, March 1988, pp 69-75.
3. R.T. Davis, M. Barnett, "The Calculation of Supersonic Viscous Flows using the Parabolized Navier-Stokes Equations," Computers and Fluids, Vol. 14, No. 3, pp 197-224, 1986.
4. B.A. Sakowski, D.D. Darling, R.L. Roach, A.G. van de Wall, "Evaluation and Application of the Baldwin-Lomax Turbulence Model in Two-Dimensional, Unsteady, Compressible Boundary Layers with and without Separation in Engine Inlets," AIAA 3676, 1992.
5. D.D. Darling, B.A. Sakowski, "Interface of an Uncoupled Boundary Layer Algorithm With an Inviscid Core Flow Algorithm for Unsteady Supersonic Engine Inlets," AIAA 92-3083, 1992.

APPENDIX

```

&GEOM X=.225528E-2,.247675E-1,.515367E-1,.778460E-1,.107633,.128037,.151619,
      .174448,.203611,.228309,.231856,.235408,.238954,.242498,
      .246048,.249659,
      .25352,.258018,.262899,.267638,.272238,.276733,
      .281173,.285603,.290036,.294476,.298924,.303379,.307842,.312309,
      .316781,.321258,.325737,.330218,
      .334703,.343673,.352644,.366070 &END
&AREAX AREA=1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
      1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
      1.0,1.0,1.0,1.0,1.0,1.0 &END
&TEMP T1=1982.9,1983.3,1983.5,1983.6,1983.5,1983.3,1982.7,1981.9,1980.3,
      1975.5,1973.5,1970.8,1966.,1960.4,1945.2,1917.8,1857.5,1776.7,
      1707.9,1605.9,1526.4,1468.8,1385.8,
      1325.1,1283.,1253.3,1228.5,1207.9,
      1188.9,1171.,1155.3,1142.7,1131.3,1122.4,
      1116.5,1106.3,1101.8,1098.6 &END
&PRES P1=.172115E7,.172253E7,.17236E7,.172399E7,.172346E7,.172235E7,
      .17199E7,.171615E7,.170832E7,.168388E7,.167384E7,.165993E7,.163667E7,
      .160348E7,.153294E7,.141769E7,.122549E7,
      .101028E7,.810816E6,.613865E6,.492139E6,.407487E6,
      .315702E6,.257376E6,.222281E6,.20018E6,.183118E6,.169834E6,
      .158537E6,.148288E6,.139641E6,.133107E6,.127322E6,.122975E6,
      .120144E6,.115385E6,
      .113278E6,.111714E6 &END
&MACH AM=.389E-1,.393E-1,.411E-1,.442E-1,.492E-1,.541E-1,.619E-1,.73E-1,
      .105,.192,.217,.25,.29,.354,.442,.561,.725,.919,1.13,1.3,1.48,1.6,
      1.77,1.89,1.98,2.01,2.1,2.14,2.19,2.22,2.25,2.29,2.31,2.32,2.34,
      2.36,2.37,2.374 &END
&RADIUSY RADY=1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
      1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
      1.0,1.0,1.0,1.0,1.0,1.0 &END
&BL JJJ=100,DUMX1=.001,II=25,TURB=.TRUE.,
      ENGU=.FALSE.,AXI=.TRUE.,EXT=.TRUE.,MYES=.TRUE. &END

```

HEADER OPTIONS DATA
TITLE BOUNDARY LAYER CODE INTERFACE WITH SINDA

MODEL = BL1
OUTPUT = BL2.OUT
USER1 = BL1.USR
USER2 = BL2.USR

HEADER CONTROL DATA, GLOBAL

NLOOPS = 4000
ABSZRO = -460.0
UID = ENG
ARLXCA = .01 \$DEFAULT VALUE
DRLXCA = .01 \$DEFAULT VALUE
EBALSA = .01 \$DEFAULT VALUE

HEADER USER DATA, GLOBAL

C

HEADER CARRAY DATA, BL

1=NASA\$PFSD:[AMBER.SINDA]LBL.DAT
2=TDISK\$DIR:[AMBER]MACH.OUT
3=TDISK\$DIR:[AMBER]C2.OUT

HEADER USER DATA, BL

C

C>>>>>RESERVE BTEST FOR USE IN BOUNDARY LAYER

HEADER ARRAY DATA, BL

C

1= 81.,0.1139 \$ SPECIFIC HEAT VS. TEMPERATURE
261.,0.1230 \$ UNITS: BTU/LBM/DEG. F
621.,0.1330 \$ AISI 304 S.S.
981.,0.1390
1341.,0.1459

2= 81., 8.61 \$ THERMAL CONDUCTIVITY VS. TEMPERATURE
261., 9.59 \$ UNITS: BTU/HR/FT/DEG. F
621.,11.44 \$ AISI 304 S.S.
981.,13.06
1341.,14.68

C

HEADER NODE DATA, BL

C*****AISI 304 S.S./DENSITY=493 LBM/FT**3

C*****GENERATE 25 DIFFUSION NODES TO REPRESENT THE WALL!!

SIM 8801,25,1,70.,A1,5.

C*****GENERATE SURFACE NODES!!!!!!!!!!>>>CONNECT TO THE BOUNDARY LAYER CODE

GEN -1,25,1,3000.,0.0

C*****GENERATE OTHER BOUNDARY NODES TO SIMULATE THE EFFECT OF A SIMPLE
C*****ACTIVE COOLING SYSTEM WHOSE EFFECTIVE TEMPERATURE IS 100 DEG F!!!

GEN -1101,25,1,100.0,0.0

HEADER CONDUCTOR DATA, BL

C*****CREATE CONDUCTORS IN THE WALL ALONG THE "X AXIS"
C*****FOR THIS CASE THE "X AXIS" FOLLOWS THE DIRECTION OF THE CORE FLOW
SIM 801,24,1,8801,1,8802,1,A2,6.

```
C*****CREATE CONDUCTORS THAT CONNECT THE WALL DIFFUSION NODES
C*****TO THE BOUNDARY NODES WHICH INTERFACE WITH THE BOUNDARY LAYER
      SIM 8801,25,1,8801,1,1,1,A2,.1
```

```
C*****CREATE CONDUCTORS THAT CONNECT THE WALL DIFFUSION NODES
C*****TO THE BOUNDARY NODES THAT SIMULATE ACTIVE COOLING
      SIM 88801,25,1,8801,1,1101,1,A2,.1
```

```
HEADER VARIABLES 1, BL
```

```
C*****THIS IS THE SUBROUTINE THE DOES IT ALL!!!!!!
      CALL INTERFACE('BL      ',25,1,1,'ENG',ABSZRO,
&      BTEST,UCA1,UCA2,UCA3,)
```

```
HEADER OPERATIONS DATA
```

```
C
BUILD BL1,BL
```

```
C      CALL STDSTL
```

```
•
•
•
•
•
```

The Proteus Navier-Stokes Code

Charles E. Towne
Trong T. Bui
Richard H. Cavicchi
Julianne M. Conley
Frank B. Molls
John R. Schwab

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

An effort is currently underway at NASA Lewis to develop two- and three-dimensional Navier-Stokes codes, called *Proteus*, for aerospace propulsion applications. The emphasis in the development of *Proteus* is not algorithm development or research on numerical methods, but rather the development of the code itself. The objective is to develop codes that are user-oriented, easily-modified, and well-documented. Well-proven, state-of-the-art solution algorithms are being used. Code readability, documentation (both internal and external), and validation are being emphasized. This paper is a status report on the *Proteus* development effort. The analysis and solution procedure are described briefly, and the various features in the code are summarized. The results from some of the validation cases that have been run are presented for both the two- and three-dimensional codes.

1. INTRODUCTION

Much of the effort in applied computational fluid dynamics consists of modifying an existing program for whatever geometries and flow regimes are of current interest to the researcher. Unfortunately, nearly all of the available nonproprietary programs were started as research projects with the emphasis on demonstrating the numerical algorithm rather than ease of use or ease of modification. The developers usually intend to clean up and formally document the program, but the immediate need to extend it to new geometries and flow regimes takes precedence.

The result is often a haphazard collection of poorly written code without any consistent structure. An extensively modified program may not even perform as expected under certain combinations of operating options. Each new user must invest considerable time and effort in attempting to understand the underlying structure of the program if intending to do anything more than run standard test cases with it. The user's subsequent modifications further obscure the program structure and therefore make it even more difficult for others to understand.

The *Proteus* two- and three-dimensional Navier-Stokes computer codes are intended to be user-oriented and easily-modifiable flow analysis programs, primarily for aerospace propulsion applications. Readability, modularity, and documentation have been the primary objectives. Every subroutine contains an extensive comment section describing the purpose, input variables, output variables, and calling sequence of the subroutine. With just three clearly-defined exceptions, the entire program is written in ANSI standard Fortran 77 to enhance portability. A master version of the program is maintained and periodically updated with corrections, as well as extensions of general interest, such as turbulence models.

The documentation is divided into three volumes. Volume 1 is the Analysis Description, and presents the equations and solution procedure used in *Proteus*. It describes in detail the governing equations, the turbulence models, the linearization of the equations and boundary conditions, the time and space differencing formulas, the ADI solution procedure, and the artificial viscosity models. Volume 2 is the User's Guide, and contains information needed to run the program. It describes the program's general features, the input and output, the procedure for setting up initial conditions, the computer resource requirements, the diagnostic messages that may be generated, the job control language used to run the program, and several test cases. Volume 3 is the Programmer's Reference, and contains detailed information useful when modifying the program. It describes the program structure, the Fortran variables stored in common blocks, and the details of each subprogram.

In this paper, the analysis and solution procedure are described briefly, and the various features in the code are summarized. The results from some of the validation cases that have been run are presented for both the two- and three-dimensional codes. The paper concludes with a brief status report on the *Proteus* development effort, including the work currently underway and our future plans.

2. ANALYSIS DESCRIPTION

In this section, the governing equations, the numerical solution method, and the turbulence models are described briefly. For a much more detailed description, see Volume 1 of the documentation (Towne, Schwab, Benson, and Suresh, 1990).

2.1 GOVERNING EQUATIONS

The basic governing equations are the compressible Navier-Stokes equations. In Cartesian coordinates, the two-dimensional planar equations can be written in strong conservation law form using vector notation as ¹

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \quad (1)$$

where

$$Q = \begin{bmatrix} \rho & \rho u & \rho v & E_T \end{bmatrix}^T \quad (2a)$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E_T + p)u \end{bmatrix} \quad (2b)$$

$$F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_T + p)v \end{bmatrix} \quad (2c)$$

$$E_v = \frac{1}{Re_r} \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - \frac{1}{Pr_r} q_x \end{bmatrix} \quad (2d)$$

$$F_v = \frac{1}{Re_r} \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - \frac{1}{Pr_r} q_y \end{bmatrix} \quad (2e)$$

The shear stresses and heat fluxes are given by

1. For brevity, in most instances this paper describes the two-dimensional *Proteus* code. The extension to three dimensions is relatively straightforward. Differences between the two-dimensional and three-dimensional codes are noted where relevant.

$$\begin{aligned}
\tau_{xx} &= 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\
\tau_{yy} &= 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\
\tau_{xy} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\
q_x &= -k \frac{\partial T}{\partial x} \\
q_y &= -k \frac{\partial T}{\partial y}
\end{aligned} \tag{3}$$

In these equations, t represents time; x and y represent the Cartesian coordinate directions; u and v are the velocities in the x and y directions; ρ , p , and T are the static density, pressure, and temperature; E_T is the total energy per unit volume; and μ , λ , and k are the coefficient of viscosity, the second coefficient of viscosity, and the coefficient of thermal conductivity.

In addition to the equations presented above, an equation of state is required to relate pressure to the dependent variables. The equation currently built into the *Proteus* code is the equation of state for thermally perfect gases, $p = \rho RT$, where R is the gas constant. For calorically perfect gases, this can be rewritten as

$$p = (\gamma - 1) \left[E_T - \frac{1}{2} \rho (u^2 + v^2) \right] \tag{4}$$

where γ is the ratio of specific heats, c_p/c_v . Additional equations are also used to define μ , λ , k , and c_p in terms of temperature for the fluid under consideration.

All of the equations have been nondimensionalized using appropriate normalizing conditions. Lengths have been nondimensionalized by L_r , velocities by u_r , density by ρ_r , temperature by T_r , viscosity by μ_r , thermal conductivity by k_r , pressure and total energy by $\rho_r u_r^2$, time by L_r/u_r , and gas constant and specific heat by u_r^2/T_r . The reference Reynolds and Prandtl numbers are thus defined as $Re_r = \rho_r u_r L_r / \mu_r$ and $Pr_r = \mu_r u_r^2 / k_r T_r$.

Because the governing equations are written in Cartesian coordinates, they are not well suited for general geometric configurations. For most applications a body-fitted coordinate system is desired. This greatly simplifies the application of boundary conditions and the bookkeeping in the numerical method used to solve the equations. The equations are thus transformed from physical (x, y, t) coordinates to rectangular orthogonal computational (ξ, η, τ) coordinates. Equation (1) becomes

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \frac{\partial \hat{E}_v}{\partial \xi} + \frac{\partial \hat{F}_v}{\partial \eta} \tag{5}$$

where

$$\hat{Q} = \frac{Q}{J}$$

$$\hat{E} = \frac{1}{J} (E \xi_x + F \xi_y + Q \xi_t)$$

$$\hat{\mathbf{F}} = \frac{1}{J}(\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{Q}\eta_t)$$

$$\hat{\mathbf{E}}_v = \frac{1}{J}(\mathbf{E}_v\xi_x + \mathbf{F}_v\xi_y)$$

$$\hat{\mathbf{F}}_v = \frac{1}{J}(\mathbf{E}_v\eta_x + \mathbf{F}_v\eta_y)$$

In these equations the derivatives ξ_x , η_x , etc., are the metric scale coefficients for the generalized nonorthogonal grid transformation. J is the Jacobian of the transformation.

2.2 NUMERICAL METHOD

2.2.1 Time Differencing. The governing equations are solved by marching in time from some known set of initial conditions using a finite difference technique. The time differencing scheme currently used is the generalized method of Beam and Warming (1978). With this scheme, the time derivative term in equation (5) is written as

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} = \frac{\Delta \hat{\mathbf{Q}}^n}{\Delta \tau} = \frac{\theta_1}{1+\theta_2} \frac{\partial(\Delta \hat{\mathbf{Q}}^n)}{\partial \tau} + \frac{1}{1+\theta_2} \frac{\partial \hat{\mathbf{Q}}^n}{\partial \tau} + \frac{\theta_2}{1+\theta_2} \frac{\Delta \hat{\mathbf{Q}}^{n-1}}{\Delta \tau} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)\Delta \tau, (\Delta \tau)^2\right] \quad (6)$$

where $\Delta \hat{\mathbf{Q}}^n = \hat{\mathbf{Q}}^{n+1} - \hat{\mathbf{Q}}^n$. The superscripts n and $n+1$ denote the known and unknown time levels, respectively. By choosing appropriate values for θ_1 and θ_2 , the solution procedure can be either first- or second-order accurate in time.

Solving equation (5) for $\partial \hat{\mathbf{Q}} / \partial \tau$, substituting the result into equation (6) for $\partial(\Delta \hat{\mathbf{Q}}^n) / \partial \tau$ and $\partial \hat{\mathbf{Q}}^n / \partial \tau$, and multiplying by $\Delta \tau$ yields

$$\begin{aligned} \Delta \hat{\mathbf{Q}}^n = & -\frac{\theta_1 \Delta \tau}{1+\theta_2} \left[\frac{\partial(\Delta \hat{\mathbf{E}}^n)}{\partial \xi} + \frac{\partial(\Delta \hat{\mathbf{F}}^n)}{\partial \eta} \right] - \frac{\Delta \tau}{1+\theta_2} \left[\frac{\partial \hat{\mathbf{E}}^n}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}^n}{\partial \eta} \right] + \frac{\theta_1 \Delta \tau}{1+\theta_2} \left[\frac{\partial(\Delta \hat{\mathbf{E}}_v^n)}{\partial \xi} + \frac{\partial(\Delta \hat{\mathbf{F}}_v^n)}{\partial \eta} \right] + \frac{\Delta \tau}{1+\theta_2} \left[\frac{\partial \hat{\mathbf{E}}_v^n}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}_v^n}{\partial \eta} \right] \\ & + \frac{\theta_2}{1+\theta_2} \Delta \hat{\mathbf{Q}}^{n-1} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta \tau)^2, (\Delta \tau)^3\right] \end{aligned} \quad (7)$$

2.2.2 Linearization Procedure. Equation (7) is nonlinear, since, for example, $\Delta \hat{\mathbf{E}}^n = \hat{\mathbf{E}}^{n+1} - \hat{\mathbf{E}}^n$ and the unknown $\hat{\mathbf{E}}^{n+1}$ is a nonlinear function of the dependent variables and of the metric coefficients resulting from the generalized grid transformation. The equations must therefore be linearized to be solved by the finite difference procedure. For the inviscid terms, and for the non-cross-derivative viscous terms, this is done by expanding each nonlinear expression in a Taylor series in time about the known time level n . The cross-derivative viscous terms are simply lagged (i.e., evaluated at the known time level n and treated as source terms.)

The linearized form of equation (7) may be written as

$$\begin{aligned}
\Delta \hat{Q}^n + \frac{\theta_1 \Delta \tau}{1 + \theta_2} \left\{ \frac{\partial}{\partial \xi} \left[\left(\frac{\partial \hat{E}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] + \frac{\partial}{\partial \eta} \left[\left(\frac{\partial \hat{F}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] \right\} - \frac{\theta_1 \Delta \tau}{1 + \theta_2} \left\{ \frac{\partial}{\partial \xi} \left[\left(\frac{\partial \hat{E}_{v_1}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] + \frac{\partial}{\partial \eta} \left[\left(\frac{\partial \hat{F}_{v_1}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] \right\} = \\
- \frac{\Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} \right)^n + \frac{\Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_1}}{\partial \xi} + \frac{\partial \hat{F}_{v_1}}{\partial \eta} \right)^n + \frac{(1 + \theta_3) \Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_2}}{\partial \xi} + \frac{\partial \hat{F}_{v_2}}{\partial \eta} \right)^n - \frac{\theta_3 \Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_2}}{\partial \xi} + \frac{\partial \hat{F}_{v_2}}{\partial \eta} \right)^{n-1} \\
+ \frac{\theta_2}{1 + \theta_2} \Delta \hat{Q}^{n-1} + O \left[\left(\theta_1 - \frac{1}{2} - \theta_2 \right) (\Delta \tau)^2, (\theta_3 - \theta_1) (\Delta \tau)^2, (\Delta \tau)^3 \right] \quad (8)
\end{aligned}$$

where $\partial \hat{E} / \partial \hat{Q}$ and $\partial \hat{F} / \partial \hat{Q}$ are the Jacobian coefficient matrices resulting from the linearization of the convective terms, and $\partial \hat{E}_{v_1} / \partial \hat{Q}$ and $\partial \hat{F}_{v_1} / \partial \hat{Q}$ are the Jacobian coefficient matrices resulting from the linearization of the viscous terms.

The boundary conditions are treated implicitly, and may be viewed simply as additional equations to be solved by the ADI solution algorithm. In general, they also involve nonlinear functions of the dependent variables. They are therefore linearized using the same procedure as for the governing equations.

2.2.3 Solution Procedure. The governing equations, presented in linearized matrix form as equation (8), are solved by an alternating direction implicit (ADI) method. The form of the ADI splitting is the same as used by Briley and McDonald (1977), and by Beam and Warming (1978). Using approximate factorization, equation (8) can be split into the following two-sweep sequence.

Sweep 1 (ξ direction)

$$\begin{aligned}
\Delta \hat{Q}^* + \frac{\theta_1 \Delta \tau}{1 + \theta_2} \frac{\partial}{\partial \xi} \left[\left(\frac{\partial \hat{E}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^* \right] - \frac{\theta_1 \Delta \tau}{1 + \theta_2} \frac{\partial}{\partial \xi} \left[\left(\frac{\partial \hat{E}_{v_1}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^* \right] = - \frac{\Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} \right)^n + \frac{\Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_1}}{\partial \xi} + \frac{\partial \hat{F}_{v_1}}{\partial \eta} \right)^n \\
+ \frac{(1 + \theta_3) \Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_2}}{\partial \xi} + \frac{\partial \hat{F}_{v_2}}{\partial \eta} \right)^n - \frac{\theta_3 \Delta \tau}{1 + \theta_2} \left(\frac{\partial \hat{E}_{v_2}}{\partial \xi} + \frac{\partial \hat{F}_{v_2}}{\partial \eta} \right)^{n-1} + \frac{\theta_2}{1 + \theta_2} \Delta \hat{Q}^{n-1} \quad (9a)
\end{aligned}$$

Sweep 2 (η direction)

$$\Delta \hat{Q}^n + \frac{\theta_1 \Delta \tau}{1 + \theta_2} \frac{\partial}{\partial \eta} \left[\left(\frac{\partial \hat{F}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] - \frac{\theta_1 \Delta \tau}{1 + \theta_2} \frac{\partial}{\partial \eta} \left[\left(\frac{\partial \hat{F}_{v_1}}{\partial \hat{Q}} \right)^n \Delta \hat{Q}^n \right] = \Delta \hat{Q}^* \quad (9b)$$

These equations represent the two-sweep alternating direction implicit (ADI) algorithm used to advance the solution from time level n to $n + 1$. \hat{Q}^* is the intermediate solution.

Spatial derivatives in equations (9a) and (9b) are approximated using second-order central difference formulas. The resulting set of algebraic equations can be written in matrix form with a block tri-diagonal coefficient matrix. They are solved using the block matrix version of the Thomas algorithm (e.g., see Anderson, Tannehill, and Pletcher, 1984).

2.2.4 Artificial Viscosity. With the numerical algorithm described above, high frequency nonlinear instabilities can appear as the solution develops. For example, in high Reynolds number flows oscillations can result from the odd-even decoupling inherent in the use of second-order central differencing for the inviscid terms. In addition, physical phenomena such as shock waves can cause instabilities when they are captured by the finite difference algorithm. Artificial viscosity, or smoothing, is normally added to the solution algorithm to suppress these high

frequency instabilities. Two artificial viscosity models are currently available in the *Proteus* computer code — a constant coefficient model used by Steger (1978), and the nonlinear coefficient model of Jameson, Schmidt, and Turkel (1981). The implementation of these models in generalized nonorthogonal coordinates is described by Pulliam (1986).

The constant coefficient model uses a combination of explicit and implicit artificial viscosity. The standard explicit smoothing uses fourth-order differences, and damps the high frequency nonlinear instabilities. Second-order explicit smoothing, while not used by Steger or Pulliam, is also available in *Proteus*. It provides more smoothing than the fourth-order smoothing, but introduces a larger error, and is therefore not used as often. The implicit smoothing is second order and is intended to extend the linear stability bound of the fourth-order explicit smoothing.

The explicit artificial viscosity is implemented in the numerical algorithm by adding the following terms to the right hand side of equation (9a) (i.e., the source term for the first ADI sweep.)

$$\frac{\varepsilon_E^{(2)} \Delta \tau}{J} (\nabla_\xi \Delta_\xi Q + \nabla_\eta \Delta_\eta Q) - \frac{\varepsilon_E^{(4)} \Delta \tau}{J} [(\nabla_\xi \Delta_\xi)^2 Q + (\nabla_\eta \Delta_\eta)^2 Q]$$

$\varepsilon_E^{(2)}$ and $\varepsilon_E^{(4)}$ are the second- and fourth-order explicit artificial viscosity coefficients. The symbols ∇ and Δ are backward and forward first difference operators.

The implicit artificial viscosity is implemented by adding the following terms to the left hand side of the equations specified.

$$- \frac{\varepsilon_I \Delta \tau}{J} [\nabla_\xi \Delta_\xi (J \Delta \hat{Q}^*)] \quad \text{to equation (9a)}$$

$$- \frac{\varepsilon_I \Delta \tau}{J} [\nabla_\eta \Delta_\eta (J \Delta \hat{Q}^*)] \quad \text{to equation (9b)}$$

The nonlinear coefficient artificial viscosity model is strictly explicit. Using the model as described by Pulliam (1986), but in the current notation, the following terms are added to the right hand side of equation (9a).

$$\nabla_\xi \left\{ \left[\left(\frac{\psi}{J} \right)_{i+1} + \left(\frac{\psi}{J} \right)_i \right] \left[\varepsilon_\xi^{(2)} \Delta_\xi Q - \varepsilon_\xi^{(4)} \Delta_\xi \nabla_\xi \Delta_\xi Q \right]_i \right\} + \nabla_\eta \left\{ \left[\left(\frac{\psi}{J} \right)_{j+1} + \left(\frac{\psi}{J} \right)_j \right] \left[\varepsilon_\eta^{(2)} \Delta_\eta Q - \varepsilon_\eta^{(4)} \Delta_\eta \nabla_\eta \Delta_\eta Q \right]_j \right\}$$

The subscripts i and j denote grid indices in the ξ and η directions. In the above expression, ψ is defined as

$$\psi = \psi_x + \psi_y$$

where ψ_x and ψ_y are spectral radii defined by

$$\psi_x = \frac{|U| + a \sqrt{\xi_x^2 + \xi_y^2}}{\Delta \xi}$$

$$\psi_y = \frac{|V| + a \sqrt{\eta_x^2 + \eta_y^2}}{\Delta \eta}$$

Here U and V are the contravariant velocities without metric normalization, defined by

$$U = \xi_t + \xi_x u + \xi_y v$$

$$V = \eta_t + \eta_x u + \eta_y v$$

and $a = \sqrt{\gamma RT}$, the speed of sound.

The parameters $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are the second- and fourth-order artificial viscosity coefficients. For the coefficients of the ξ direction differences,

$$\left[\epsilon_{\xi}^{(2)}\right]_i = \kappa_2 \Delta \tau \max(\sigma_{i+1}, \sigma_i, \sigma_{i-1})$$

$$\left[\epsilon_{\xi}^{(4)}\right]_i = \max\left[0, \kappa_4 \Delta \tau - \left[\epsilon_{\xi}^{(2)}\right]_i\right]$$

where

$$\sigma_i = \left| \frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}} \right|$$

and κ_2 and κ_4 are constants. Similar formulas are used for the coefficients of the η direction differences. The parameter σ is a pressure gradient scaling parameter that increases the amount of second-order smoothing relative to fourth-order smoothing near shock waves. The logic used to compute $\epsilon^{(4)}$ switches off the fourth-order smoothing when the second-order smoothing term is large.

2.3 TURBULENCE MODELS

Turbulence is modeled using either a generalized version of the Baldwin and Lomax (1978) algebraic eddy viscosity model, or the Chien (1982) low Reynolds number k - ϵ model.

2.3.1 Baldwin-Lomax Model. For wall-bounded flows, the Baldwin-Lomax turbulence model is a two-layer model, with

$$\mu_t = \begin{cases} (\mu_t)_{inner} & \text{for } y_n \leq y_b \\ (\mu_t)_{outer} & \text{for } y_n > y_b \end{cases} \quad (10)$$

where y_n is the normal distance from the wall, and y_b is the smallest value of y_n at which the values of μ_t from the inner and outer region formulas are equal. For free turbulent flows, only the outer region value is used.

The outer region turbulent viscosity at a given ξ or η station is computed from

$$(\mu_t)_{outer} = KC_{cp} \rho F_{Kleb} F_{wake} Re_r \quad (11)$$

where K is the Clauser constant, taken as 0.0168, and C_{cp} is a constant taken as 1.6.

The parameter F_{wake} is computed from

$$F_{wake} = \begin{cases} y_{max} F_{max} & \text{for wall-bounded flows} \\ C_{wt} V_{diff}^2 \frac{y_{max}}{F_{max}} & \text{for free turbulent flows} \end{cases} \quad (12)$$

where C_{wt} is a constant taken as 0.25, and

$$V_{diff} = |\vec{V}|_{max} - |\vec{V}|_{min}$$

where \vec{V} is the total velocity vector.

The parameter F_{max} in equation (12) is the maximum value of

$$F(y_n) = \begin{cases} y_n |\vec{\Omega}| \left(1 - e^{-y^*/A^+}\right) & \text{for wall-bounded flows} \\ y_n |\vec{\Omega}| & \text{for free turbulent flows} \end{cases} \quad (13)$$

and y_{max} is the value of y_n corresponding to F_{max} .

For wall-bounded flows, y_n is the normal distance from the wall. For free turbulent flows, two values of F_{max} and y_{max} are computed — one using the location of $|\vec{V}|_{max}$ as the origin for y_n , and one using the location of $|\vec{V}|_{min}$. The origin giving the smaller value of y_{max} is the one finally used for computing y_n , F_{max} , and y_{max} .

In equation (13), $|\vec{\Omega}|$ is the magnitude of the total vorticity, defined for two-dimensional planar flow as

$$|\vec{\Omega}| = \left| \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right| \quad (14)$$

The parameter A^+ is the Van Driest damping constant, taken as 26.0. The coordinate y^+ is defined as

$$y^+ = \frac{\rho_w u_\tau y_n}{\mu_w} Re_\tau = \frac{\sqrt{\tau_w \rho_w} Re_\tau}{\mu_w} y_n \quad (15)$$

where $u_\tau = \sqrt{\tau_w / \rho_w Re_\tau}$ is the friction velocity, τ is the shear stress, and the subscript w indicates a wall value. In *Proteus*, τ_w is set equal to $\mu_w |\vec{\Omega}|_w$.

The function F_{Kleb} in equation (11) is the Klebanoff intermittency factor. For free turbulent flows, $F_{Kleb} = 1$. For wall-bounded flows,

$$F_{Kleb} = \left[1 + B \left(\frac{C_{Kleb} y_n}{y_{max}} \right)^6 \right]^{-1} \quad (16)$$

In equation (16), B and C_{Kleb} are constants taken as 5.5 and 0.3, respectively.

The inner region turbulent viscosity in the Baldwin-Lomax model is

$$(\mu_t)_{inner} = \rho l^2 |\vec{\Omega}| Re_\tau \quad (17)$$

where l is the mixing length, given by

$$l = \kappa y_n \left(1 - e^{-y^*/A^+} \right) \quad (18)$$

and κ is the Von Karman constant, taken as 0.4.

If both boundaries in a given coordinate direction are solid surfaces, the turbulence model is applied separately for each surface. An averaging procedure is used to combine the resulting two μ_t profiles into one.

The turbulent second coefficient of viscosity is simply defined as

$$\lambda_t = -\frac{2}{3} \mu_t$$

The turbulent thermal conductivity coefficient is defined using Reynolds analogy as

$$k_t = \frac{c_p \mu_t}{Pr_t} Pr_t$$

where c_p is the specific heat at constant pressure, and Pr_t is the turbulent Prandtl number.

2.3.2 Chien k - ϵ Model. The low Reynolds number k - ϵ formulation of Chien (1982) was chosen because of its reasonable approximation of the near wall region and because of its numerical stability. Here k and ϵ are the turbulent kinetic energy and the turbulent dissipation rate, respectively.

In Cartesian coordinates, the two-dimensional planar equations for the Chien k - ϵ model can be written using vector notation as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} + \mathbf{T} \quad (19)$$

where

$$\mathbf{W} = \begin{bmatrix} \rho k \\ \rho \epsilon \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} \rho u k - \frac{1}{Re_r} \mu_k \frac{\partial k}{\partial x} \\ \rho u \epsilon - \frac{1}{Re_r} \mu_\epsilon \frac{\partial \epsilon}{\partial x} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \rho v k - \frac{1}{Re_r} \mu_k \frac{\partial k}{\partial y} \\ \rho v \epsilon - \frac{1}{Re_r} \mu_\epsilon \frac{\partial \epsilon}{\partial y} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} P_k - Re_r \rho \epsilon \\ C_1 P_k \frac{\epsilon}{k} - Re_r C_2 \rho \frac{\epsilon^2}{k} \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} -\frac{2}{Re_r} \frac{\mu}{y_n^2} k \\ -\frac{2}{Re_r} \frac{\mu \epsilon^{-y^*/2}}{y_n^2} \epsilon \end{bmatrix}$$

and

$$\mu_k = \mu + \frac{\mu_t}{\sigma_k}$$

$$\mu_\epsilon = \mu + \frac{\mu_t}{\sigma_\epsilon}$$

$$C_2 = C_2 \left[1 - \frac{2}{9} e^{-R_t^2/36} \right]$$

$$R_t = \frac{\rho k^2}{\mu \epsilon}$$

$$P_k = \frac{\mu_t}{Re_r} P_1 - \frac{2}{3} \rho k P_2$$

$$P_1 = 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2$$

$$P_2 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

The turbulent viscosity is given by

$$\mu_t = C_\mu \rho \frac{k^2}{\varepsilon} \quad (20)$$

$$C_\mu = C_{\mu, \infty} \left(1 - e^{-C_3 y^+} \right)$$

In the above equations, C_1 , C_2 , C_3 , σ_k , σ_ε , and C_μ , are constants equal to 1.35, 1.8, 0.0115, 1.0, 1.3, and 0.09, respectively. The parameter y_n is the minimum distance to the nearest solid surface, and y^+ is computed from y_n . In the above equations the mean flow properties have been nondimensionalized as described in Section 2.1. The turbulent kinetic energy k and the turbulent dissipation rate ε have been nondimensionalized by u_τ^2 and $\rho_\tau u_\tau^4 / \mu_\tau$, respectively.

After transforming from physical to rectangular orthogonal computational coordinates, equation (19) becomes

$$\frac{\partial \hat{W}}{\partial \tau} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = \hat{S} + \hat{T} \quad (21)$$

where

$$\hat{W} = \frac{1}{J} \left[\frac{\rho k}{\rho \varepsilon} \right]$$

$$\hat{F} = \hat{F}_C - \hat{F}_D - \hat{F}_M$$

$$\hat{F}_C = \frac{1}{J} \left[\frac{\xi_x \rho u k + \xi_y \rho v k}{\xi_x \rho u \varepsilon + \xi_y \rho v \varepsilon} \right]$$

$$\hat{F}_D = \frac{1}{J} \frac{1}{Re_\tau} \left[\frac{\mu_k (\xi_x^2 + \xi_y^2) k_\xi}{\mu_\varepsilon (\xi_x^2 + \xi_y^2) \varepsilon_\xi} \right]$$

$$\hat{F}_M = \frac{1}{J} \frac{1}{Re_\tau} \left[\frac{\mu_k (\xi_x \eta_x + \xi_y \eta_y) k_\eta}{\mu_\varepsilon (\xi_x \eta_x + \xi_y \eta_y) \varepsilon_\eta} \right]$$

$$\hat{G} = \hat{G}_C - \hat{G}_D - \hat{G}_M$$

$$\hat{G}_C = \frac{1}{J} \left[\frac{\eta_x \rho u k + \eta_y \rho v k}{\eta_x \rho u \varepsilon + \eta_y \rho v \varepsilon} \right]$$

$$\hat{G}_D = \frac{1}{J} \frac{1}{Re_\tau} \left[\frac{\mu_k (\eta_x^2 + \eta_y^2) k_\eta}{\mu_\varepsilon (\eta_x^2 + \eta_y^2) \varepsilon_\eta} \right]$$

$$\hat{G}_M = \frac{1}{J} \frac{1}{Re_r} \left[\frac{\mu_k(\xi_x \eta_x + \xi_y \eta_y) k_\xi}{\mu_\epsilon(\xi_x \eta_x + \xi_y \eta_y) \epsilon_\xi} \right]$$

$$\hat{S} = \frac{1}{J} \left[\frac{P_k - Re_r \rho \epsilon}{C_1 P_k \frac{\epsilon}{k} - Re_r C_2 \rho \frac{\epsilon^2}{k}} \right]$$

$$\hat{T} = \frac{1}{J} \left[\frac{-\frac{2}{Re_r} \frac{\mu}{y_n^2} k}{-\frac{2}{Re_r} \frac{\mu e^{-\gamma^*/2}}{y_n^2} \epsilon} \right]$$

The time differencing scheme and linearization procedure described previously for the mean flow equations are also applied to equation (21). The mean flow variables are evaluated at the known time level n . This allows the k - ϵ equations to be uncoupled from the mean flow equations and solved separately. Spatial derivatives are approximated using first-order upwind differences for the convective terms, and second-order central differences for the viscous terms. In the two-dimensional *Proteus* code, the equations are solved by the same ADI procedure as the mean flow equations. In the three-dimensional code, they are solved by a two-sweep LU procedure, as described by Hoffmann (1989).

The turbulent second coefficient of viscosity λ , and the turbulent thermal conductivity coefficient k , are defined as described in the previous section.

3. CODE FEATURES

In this section the basic characteristics and capabilities of the two- and three-dimensional *Proteus* codes are summarized. For a much more detailed description, see Volumes 2 and 3 of the documentation (Towne, Schwab, Benson, and Suresh, 1990).

3.1 ANALYSIS

The *Proteus* codes solve the unsteady compressible Navier-Stokes equations in either two or three dimensions. The 2-D code can solve either the planar or axisymmetric form of the equations. Swirl is allowed in axisymmetric flow. The 2-D planar equations and the 3-D equations are solved in fully conservative form. As subsets of these equations, options are available to solve the Euler equations or the thin-layer Navier-Stokes equations. An option is also available to eliminate the energy equation by assuming constant total enthalpy.

The equations are solved by marching in time using the generalized time differencing of Beam and Warming (1978). The method may be either first- or second-order accurate in time, depending on the choice of time differencing parameters. Second-order central differencing is used for all spatial derivatives. Nonlinear terms are linearized using second-order Taylor series expansions in time. The resulting difference equations are solved using an alternating-direction implicit (ADI) technique, with Douglas-Gunn type splitting as written by Briley and McDonald (1977). The boundary conditions are also treated implicitly.

Artificial viscosity, or smoothing, is normally added to the solution algorithm to damp pre- and post-shock oscillations in supersonic flow, and to prevent odd-even decoupling due to the use of central differences in convection-dominated regions of the flow. Implicit smoothing and two types of explicit smoothing are available in *Proteus*. The implicit smoothing is second order with constant coefficients. For the explicit smoothing the user may choose a constant coefficient second- and/or fourth-order model (Steger, 1978), or a nonlinear coefficient mixed second- and fourth-order model (Jameson, Schmidt, and Turkel, 1981). The nonlinear coefficient model was designed specifically for flow with shock waves.

The equations are fully coupled, leading to a system of equations with a block tridiagonal coefficient matrix that can be solved using the block matrix version of the Thomas algorithm. Because this algorithm is recursive, the source code cannot be vectorized in the ADI sweep direction. However, it is vectorized in the non-sweep direction,

leading to an efficient implementation of the algorithm.

3.2 GEOMETRY AND GRID SYSTEM

The equations solved in *Proteus* were originally written in a Cartesian coordinate system, then transformed into a general nonorthogonal computational coordinate system. The code is therefore not limited to any particular type of geometry or coordinate system. The only requirement is that body-fitted coordinates must be used. In general, the computational coordinate system for a particular geometry must be created by a separate coordinate generation code and stored in an unformatted file that *Proteus* can read. However, simple Cartesian and polar coordinate systems are built in.

The equations are solved at grid points that form a computational mesh within this computational coordinate system. The number of grid points in each direction in the computational mesh is specified by the user. The location of these grid points can be varied by packing them at either or both boundaries in any coordinate direction. The transformation metrics and Jacobian are computed using finite differences in a manner consistent with the differencing of the governing equations.

3.3 FLOW AND REFERENCE CONDITIONS

As stated earlier, the equations solved by *Proteus* are for compressible flow. Incompressible conditions can be simulated by running at a Mach number of around 0.1. Lower Mach numbers may lead to numerical problems. The flow can be laminar or turbulent. The gas constant \bar{R} is specified by the user, with the value for air as the default. The specific heats c_p and c_v , the molecular viscosity μ , and the thermal conductivity k can be treated as constants or as functions of temperature. The empirical formulas used to relate these properties to temperature are contained in a separate subroutine, and can easily be modified if necessary. The perfect gas equation of state is used to relate pressure, density, and temperature. This equation is also contained in a separate subroutine, which could be easily modified if necessary. All equations and variables in the program are nondimensionalized by normalizing values derived from reference conditions specified by the user, with values for sea level air as the default.

3.4 BOUNDARY CONDITIONS

The easiest way to specify boundary conditions in *Proteus* is by specifying the type of boundary (e.g., no-slip adiabatic wall, subsonic inflow, periodic, etc.). The program will then select an appropriate set of conditions for that boundary. For many applications this method should be sufficient. If necessary, however, the user may instead set the individual boundary conditions on any or all of the computational boundaries.

A variety of individual boundary conditions are built into the *Proteus* code, including: (1) specified values and/or gradients of Cartesian velocities u , v , and w , normal and tangential velocities V_n and V_t , pressure p , temperature T , and density ρ ; (2) specified values of total pressure p_T , total temperature T_T , and flow angle; and (3) linear extrapolation. Another useful boundary condition is a "no change from initial condition" option for u , v , w , p , T , ρ , p_T , and/or T_T . Provision is also made for user-written boundary conditions. Specified gradient boundary conditions may be in the direction of the coordinate line intersecting the boundary or normal to the boundary, and may be computed using two-point or three-point difference formulas. For all of these conditions, the same type and value may be applied over the entire boundary surface, or a point-by-point distribution may be specified. Unsteady and time-periodic boundary conditions are allowed when applied over the entire boundary.

3.5 INITIAL CONDITIONS

Initial conditions are required throughout the flow field to start the time marching procedure. For unsteady flows they should represent a real flow field. A converged steady-state solution from a previous run would be a good choice. For steady flows, the ideal initial conditions would represent a real flow field that is close to the expected final solution.

The best choice for initial conditions, therefore, will vary from problem to problem. For this reason *Proteus* does not include a general-purpose routine for setting up initial conditions. The user must supply a subroutine, called INIT, that sets up the initial starting conditions for the time marching procedure. A version of INIT is, however, built into *Proteus* that specifies uniform flow with constant flow properties everywhere in the flow field. These conditions, of course, do represent a solution to the governing equations, and for many problems may help minimize starting transients in the time marching procedure. However, realistic initial conditions that are closer to the expected final solution should lead to quicker convergence.

3.6 TIME STEP SELECTION

Several different options are available for choosing the time step $\Delta\tau$, and for modifying it as the solution proceeds. $\Delta\tau$ may be specified directly, or through a value of the Courant-Friedrichs-Lewy (CFL) number. When specifying a CFL number, the time step may be either *global* (i.e., constant in space) based on the minimum CFL limit, or *local* (i.e., varying in space) based on the local CFL limit. For unsteady time-accurate flows global values should be used, but for steady flows using local values may lead to faster convergence. Options are available to increase or decrease $\Delta\tau$ as the solution proceeds based on the change in the dependent variables. An option is also available to cycle $\Delta\tau$ between two values in a logarithmic progression over a specified number of time steps.

3.7 CONVERGENCE

Five options are currently available for determining convergence. The user specifies a convergence criterion ϵ for each of the governing equations. Then, depending on the option chosen, convergence is based on: (1) the absolute value of the maximum change in the conservation variables ΔQ_{\max} over a single time step; (2) the absolute value of the maximum change ΔQ_{\max} averaged over a specified number of time steps; (3) the L_2 norm of the residual for each equation; (4) the average residual for each equation; or (5) the maximum residual for each equation.

It should be noted, however, that convergence is in the eye of the beholder. The amount of decrease in the residual necessary for convergence will vary from problem to problem. For some problems, it may be more appropriate to measure convergence by some flow-related parameter, such as the lift coefficient for an airfoil. Determining when a solution is sufficiently converged is, in some respects, a skill best acquired through experience.

3.8 INPUT/OUTPUT

Input to *Proteus* is through a series of namelists and, in general, an unformatted file containing the computational coordinate system. All of the input parameters have default values and only need to be specified by the user if a different value is desired. Reference conditions may be specified in either English or SI units. A restart option is also available, in which the computational mesh and the initial flow field are read from unformatted restart files created during an earlier run.

The standard printed output available in *Proteus* includes an echo of the input, boundary conditions, normalizing and reference conditions, the computed flow field, and convergence information. The user controls exactly which flow field parameters are printed, and at which time levels and grid points. Several debug options are also available for detailed printout in various parts of the program.

In addition to the printed output, several unformatted files can be written for various purposes. The first is an auxiliary file used for post-processing, usually called a plot file, that can be written at convergence or after the last time step if the solution does not converge. Plot files can be written for the NASA Lewis plotting program *CONTOUR* or the NASA Ames plotting program *PLOT3D*. If *PLOT3D* is to be used, two unformatted files are created, an *xyz* file containing the computational mesh and a *q* file containing the computed flow field. Another unformatted file written by *Proteus* contains detailed convergence information. This file is automatically incremented each time the solution is checked for convergence, and is used to generate the convergence history printout and with Lewis-developed post-processing plotting routines. And finally, two unformatted files may be written at the end of a calculation that may be used to restart the calculation in a later run. One of these contains the computational mesh, and the other the computed flow field.

3.9 TURBULENCE MODELS

For turbulent flow, *Proteus* solves the Reynolds time-averaged Navier-Stokes equations, with turbulence modeled using either the Baldwin and Lomax (1978) algebraic eddy-viscosity model or the Chien (1982) two-equation model.

3.9.1 Baldwin-Lomax Model. The Baldwin-Lomax model may be applied to either wall-bounded flows or to free turbulent flows. For wall-bounded flows, the model is a two-layer model. For flows in which more than one boundary is a solid surface, averaging procedures are used to determine a single μ_t profile. The turbulent thermal conductivity coefficient k_t is computed using Reynolds analogy.

3.9.2 Chien k - ϵ Model. With the Chien two-equation model, partial differential equations are solved for the turbulent kinetic energy k and the turbulent dissipation rate ϵ . These equations are lagged in time and solved separately from the mean flow equations. In the 2-D *Proteus* code, the equations are solved using the same solution

algorithm as for the mean flow equations, except that spatial derivatives for the convective terms are approximated using first-order upwind differencing. In the 3-D code, they are solved by a two-sweep LU procedure, as described by Hoffmann (1989).

Since the Chien two-equation model is a low Reynolds number formulation, the k - ϵ equations are solved in the near-wall region. No additional approximations are needed. Boundary conditions that may be used include: (1) no change from initial or restart conditions for k and ϵ ; (2) specified values and/or gradients of k and ϵ ; and (3) linear extrapolation. Specified gradient boundary conditions are in the direction of the coordinate line intersecting the boundary, and may be computed using two-point or three-point difference formulas. For all of these conditions, the same type and value may be applied over the entire boundary surface, or a point-by-point distribution may be specified. Spatially periodic boundary conditions for k and ϵ may also be used. Unsteady boundary conditions are not available for the k - ϵ equations. However, unsteady flows can still be computed with the Chien model using the unsteady boundary condition option for the mean flow quantities and appropriate boundary conditions for k and ϵ , such as specified gradients or linear extrapolation.

Initial conditions for k and ϵ are required throughout the flow field to start the time marching procedure. The best choice for initial conditions will vary from problem to problem, and the user may supply a subroutine, called KEINIT, that sets up the initial values of k and ϵ for the time marching procedure. A version of KEINIT is built into *Proteus* that computes the initial values from a mean initial or restart flow field based on the assumption of local equilibrium (i.e., production equals dissipation.) Variations of that scheme have been found to be useful in computing initial k and ϵ values for a variety of turbulent flows.

The time step used in the solution of the k - ϵ equations is normally the same as the time step used for the mean flow equations. However, the user can alter the time step, making it larger or smaller than the time step for the mean flow equations, by specifying a multiplication factor. The user can also specify the number of k - ϵ iterations per mean flow iteration.

4. VERIFICATION CASES

Throughout the *Proteus* development effort, verification of the code has been emphasized. A variety of cases have been run, and the computed results have been compared with both experimental data and exact solutions. Some cases are included in Volume 2 of the *Proteus* documentation (Towne, Schwab, Benson, and Suresh, 1990). Other cases have been reported by Conley and Zeman (1991), Saunders and Keith (1991), and Bui (1992).

Three cases are presented in this paper — flow past a circular cylinder, flow through a transonic diffuser, and flow through a square-cross-sectioned S-duct.

4.1 FLOW PAST A CIRCULAR CYLINDER

In this test case, steady flow past a two-dimensional circular cylinder was investigated. Both Euler and laminar viscous flow were computed.

4.1.1 Reference Conditions. In order to allow comparison of the *Proteus* results with incompressible experimental data and with potential flow results, this case was run with a low reference Mach number of 0.2. The cylinder radius was used as the reference length, and was set equal to 1 ft. Standard sea level conditions of 519 °R and 0.07645 lb_m/ft³ were used for the reference temperature and density. The Reynolds number based on cylinder diameter was 40, matching the experimental value.

4.1.2 Computational Coordinates. For this problem a polar computational coordinate system was the obvious choice. The radial coordinate r varied from 1 at the cylinder surface to 30 at the outer boundary. Since the flow is symmetric, only the top half of the flow field was computed. The circumferential coordinate θ thus varied from 0° at the cylinder leading edge to 180° at the trailing edge. For the Euler flow case, a 21 (circumferential) \times 51 (radial) mesh was used, with the radial grid packed moderately tightly near the cylinder surface. For the viscous flow case, a 51 \times 51 mesh was used, with the radial grid packed more tightly near the cylinder surface.

4.1.3 Initial Conditions. Constant stagnation enthalpy was assumed, so only three initial conditions were required. For the Euler flow case, uniform flow with $u = 1$, $v = 0$, and $p = 1$ was used.

For the viscous flow case, the exact potential flow solution was used to set the initial conditions at all the non-wall points. Thus, with nondimensional free stream conditions of $\rho_\infty = u_\infty = T_\infty = p_\infty = 1$, the initial conditions

were²

$$u = 1 - \frac{1}{r^2} \cos(2\theta)$$

$$v = -\frac{1}{r^2} \sin(2\theta)$$

$$p = (p_T)_\infty - \frac{1}{2} \frac{\rho_\infty (u^2 + v^2)}{R}$$

where

$$(p_T)_\infty = p_\infty + \frac{1}{2} \frac{\rho_\infty u_\infty^2}{R}$$

At the cylinder surface, the initial velocities u and v were set equal to zero, and the pressure p was set equal to the pressure at the grid point adjacent to the surface. Thus, with two-point one-sided differencing, $\partial p / \partial n = 0$ at the surface.

4.1.4 Boundary Conditions. Again, since we assumed constant stagnation enthalpy, only three boundary conditions were required at each computational boundary. For the Euler flow case, symmetry conditions were used along the symmetry line ahead of and behind the cylinder. At the cylinder surface, the radial velocity and the radial gradient of the circumferential velocity were set equal to zero. The radial gradient of pressure was computed from the polar coordinate form of the incompressible radial momentum equation written at the wall. The equation is (Hughes and Gaylord, 1964)

$$\rho v_r \frac{\partial v_r}{\partial r} + \frac{\rho v_\theta}{r} \frac{\partial v_r}{\partial \theta} - \rho \frac{v_\theta^2}{r} = -\frac{\partial p}{\partial r}$$

where v_r and v_θ are the radial and circumferential velocities, respectively. At the cylinder surface, $v_r = 0$. Thus,

$$\frac{\partial p}{\partial r} = \rho \frac{v_\theta^2}{r} = \rho \frac{u^2 + v^2}{r}$$

And finally, at the outer boundary the free stream conditions were specified as boundary conditions.

For the viscous flow case, symmetry conditions were again used along the symmetry line ahead of and behind the cylinder. At the cylinder surface, no-slip conditions were used for the velocity, and the radial pressure gradient was set equal to zero. The outer boundary was split into an inlet region and wake region. The split was made, somewhat arbitrarily, at $\theta = 135^\circ$. In the inlet region, the boundary values of u , v , and p were kept at their initial values, which were the potential flow values. In the wake region, the boundary values of p were kept at their initial values, and the radial gradients of u and v were set equal to zero.

4.1.5 Numerics. Both the Euler and viscous flow cases were run using a spatially varying time step, with a local CFL number of 10. The constant coefficient artificial viscosity model was used, with $\epsilon_1 = 2$ and $\epsilon_2^{(4)} = 1$.

The Euler flow case converged in 210 time steps, and the viscous flow case converged in 360 time steps. The convergence criterion for both cases was that the L_2 norm of the residual for each equation drop below 0.001.

2. Note that the nondimensional gas constant R appears in these equations. This is because, in the *Proteus* input and output, the pressure is nondimensionalized by $\rho_\infty \bar{R} T_\infty$. Internal to the code, pressure is nondimensionalized by $\rho_\infty u_\infty^2$, as described in Section 2.1.

4.1.6 Computed Results. In Figure 1 the computed static pressure coefficient, defined as $(\bar{p} - p_\infty) / (\rho_\infty u_\infty^2 / 2g_c)$ is plotted as a function of θ for both the Euler and viscous flow cases. Also shown are the experimental data of Grove, Shair, Petersen, and Acrivos (1964), and the exact solution for potential flow. The *Proteus* results agree well with the data for the viscous flow case, and with the exact potential flow solution for the Euler flow case.

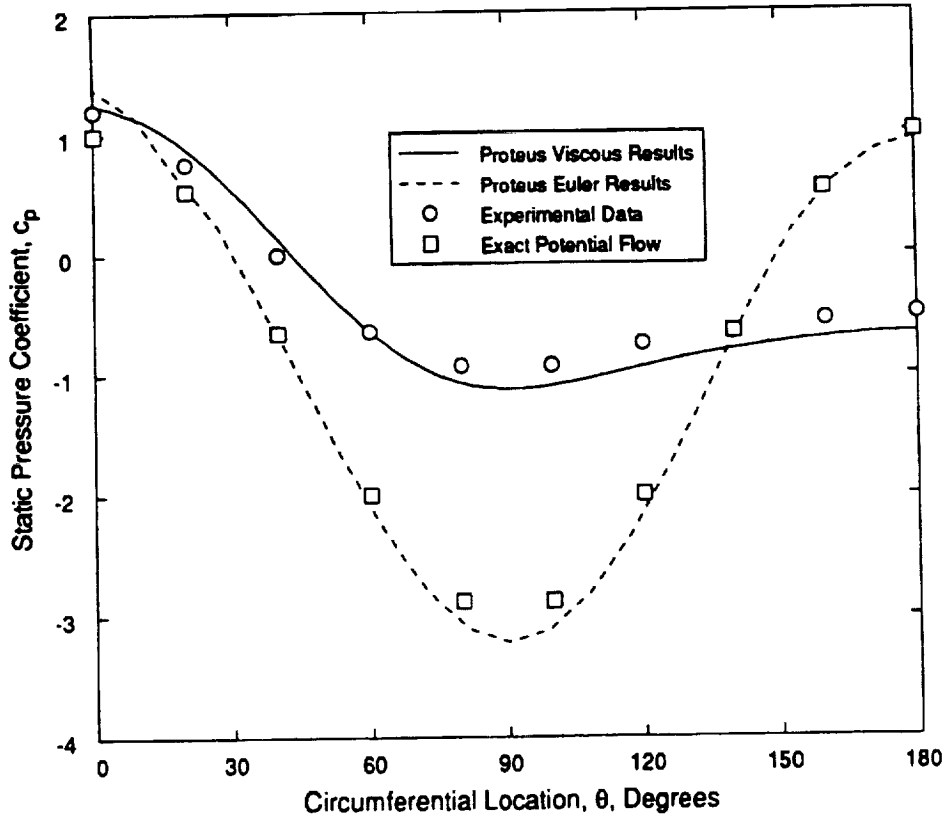


Figure 1. Pressure coefficient for flow past a circular cylinder.

4.2 TRANSONIC DIFFUSER FLOW

In this test case, two-dimensional transonic turbulent flow was computed in a converging-diverging duct. Turbulence was modeled using the Baldwin-Lomax model. The flow entered the duct subsonically, accelerated through the throat to supersonic speed, then decelerated through a normal shock and exited the duct subsonically. The computational domain is shown in Figure 2.

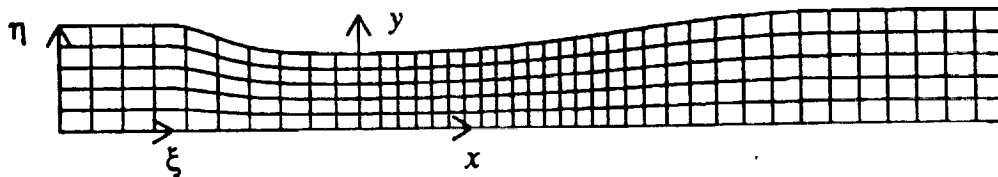


Figure 2. Computational domain for transonic diffuser flow.

4.2.1 Reference Conditions. The throat height of 0.14435 ft. was used as the reference length L_r . The reference velocity u_r was 100 ft/sec. The reference temperature and density were 525.602 °R and 0.1005 lb_m/ft³, respectively. These values match the inlet total temperature and total pressure used in other numerical simulations of this

flow (Hsieh, Bogar, and Coakley, 1987).

4.2.2 Computational Coordinates. The x coordinate for this duct runs from -4.04 to $+8.65$. The Cartesian coordinates of the bottom wall are simply $y = 0$ for all x . For the top wall, the y coordinate is given by (Bogar, Sajben, and Kroutil, 1983)

$$y = \begin{cases} 1.4144 & \text{for } x \leq -2.598 \\ \alpha \cosh \zeta / (\alpha - 1 + \cosh \zeta) & \text{for } -2.598 < x < 7.216 \\ 1.5 & \text{for } x \geq 7.216 \end{cases}$$

where the parameter ζ is defined as

$$\zeta = \frac{C_1(x/x_l)[1 + C_2x/x_l]^{C_3}}{(1 - x/x_l)^{C_4}}$$

The various constants used in the formula for the top wall height in the converging ($-2.598 \leq x \leq 0$) and diverging ($0 \leq x \leq 7.216$) parts of the duct are given in the following table.

Constant	Converging	Diverging
α	1.4114	1.5
x_l	-2.598	7.216
C_1	0.81	2.25
C_2	1.0	0.0
C_3	0.5	0.0
C_4	0.6	0.0

A body-fitted coordinate system was generated for the duct, with 81 points in the x direction and 51 points in the y direction. The coordinate system is shown in Figure 2. For clarity, the grid points are thinned by factors of 2 and 10 in the x and y directions, respectively. Note that for good resolution of the flow near the normal shock, the grid defining the computational coordinate system is denser in the x direction in the region just downstream of the throat. In the y direction, the actual computational mesh was tightly packed near both walls to resolve the turbulent boundary layers.

4.2.3 Initial Conditions. The initial conditions were simply zero velocity and constant pressure and temperature. Thus, $u = v = 0$ and $p = T = 1$ everywhere in the flow field.

4.2.4 Boundary Conditions. This calculation was performed in three separate runs. In the first run, the exit static pressure was gradually lowered to a value low enough to establish supersonic flow throughout the diverging portion of the duct. The pressure was lowered as follows:

$$p(i) = \begin{cases} 0.99 & \text{for } 1 < n \leq 100 \\ -2.1405 \times 10^{-3}n + 1.20405 & \text{for } 101 \leq n \leq 500 \\ 0.1338 & \text{for } 501 \leq n \leq 3001 \end{cases}$$

where n is the time level. The equation for p for $101 \leq n \leq 500$ is simply a linear interpolation between $p = 0.99$ and $p = 0.1338$. In the second run, the exit pressure was gradually raised to a value consistent with the formation of a normal shock just downstream of the throat. Thus,

$$p(i) = \begin{cases} 3.4327 \times 10^{-4}n - 0.89636 & \text{for } 3001 < n \leq 5000 \\ 0.82 & \text{for } 5001 \leq n \leq 6001 \end{cases}$$

Again, the equation for p for $3001 < n \leq 5000$ is simply a linear interpolation between $p = 0.1338$ and $p = 0.82$. In the third run, the exit pressure was kept constant at 0.82 for $6001 < n \leq 9000$.

The remaining boundary conditions were the same for all runs. At the inlet, the total pressure and total temperature were set equal to 1, and the y -velocity and the normal gradient of the x -velocity were both set equal to zero. At the exit, the normal gradients of temperature and both velocity components were set equal to zero. At both walls, no-slip adiabatic conditions were used, and the normal pressure gradient was set equal to zero.

4.2.5 Numerics. The case was run using a spatially varying time step. The local CFL number was 0.5 for the first two runs, and 5.0 for the third run. The nonlinear coefficient artificial viscosity model was used. For the first two runs, the coefficients $\epsilon^{(2)}$ and $\epsilon^{(4)}$ were 0.1 and 0.005, respectively. For the third run, $\epsilon^{(4)}$ was lowered to 0.0004.

The convergence criterion was that the absolute value of the maximum change in the conservation variables ΔQ_{max} be less than 10^{-6} . At the end of the third run, the solution had not yet converged to this level. However, close examination of several parameters near the end of the calculation indicates that the solution is no longer changing appreciably with time, but oscillates slightly about some mean steady level. This type of result appears to be fairly common, especially for flows with shock waves. The reason is not entirely clear, but may be related to inadequate mesh resolution, discontinuities in metric information, etc. For this particular case, the cause may also be inherent unsteadiness in the flow. The experimental data for this duct show a self-sustained oscillation of the normal shock at Mach numbers greater than about 1.3 (Bogar, Sajben, and Kroutil, 1983).

4.2.6 Computed Results. The computed flow field is shown in Figure 3 in the form of constant Mach number contours.

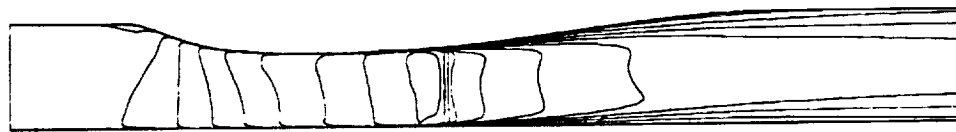


Figure 3. Computed Mach number contours for transonic diffuser flow.

The flow enters the duct at about $M = 0.46$, accelerates to just under $M = 1.3$ slightly downstream of the throat, shocks down to about $M = 0.78$, then decelerates and leaves the duct at about $M = 0.51$. The normal shock in the throat region and the growing boundary layers in the diverging section can be seen clearly. Because this is a shock capturing analysis, the normal shock is smeared in the streamwise direction.

The computed distribution of the static pressure ratio along the top and bottom walls is compared with experimental data (Hsieh, Wardlaw, Collins, and Coakley, 1987) in Figure 4. The static pressure ratio is here defined as $p / (p_T)_0$, where $(p_T)_0$ is the inlet core total pressure.

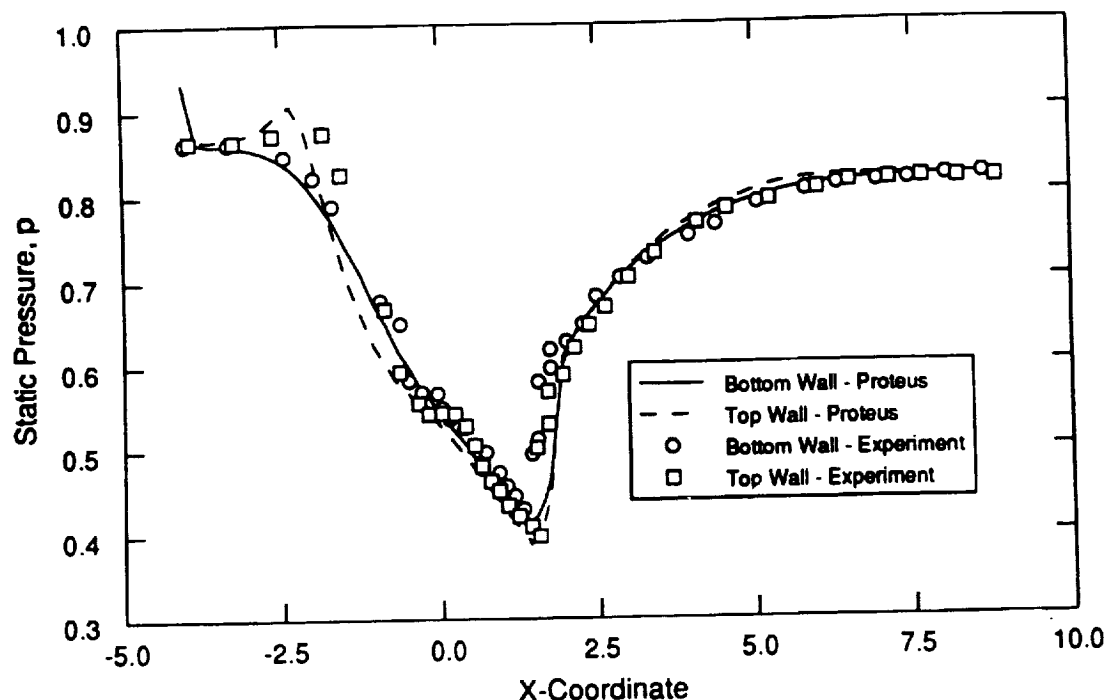


Figure 4. Computed and experimental static pressure distribution for transonic diffuser flow.

The computed results generally agree well with the experimental data, including the jump conditions across the normal shock. The predicted shock position, however, is slightly downstream of the experimentally measured position. The pressure change, of course, is also smeared over a finite distance. There is also some disagreement between analysis and experiment along the top wall near the inlet. This may be due to rapid changes in the wall contour in this region without sufficient mesh resolution.

4.3 TURBULENT S-DUCT FLOW

In this test case, three-dimensional turbulent flow in an S-duct was computed using first the Baldwin-Lomax algebraic turbulence model and then the Chien $k-\epsilon$ turbulence model. The S-duct consisted of two 22.5° bends with a constant area square cross section. The geometry and experimental data were obtained from a test conducted by Taylor, Whitelaw, and Yianneskis (1982).

4.3.1 Reference Conditions. The default standard sea level conditions for air of 519°R and $0.07645 \text{ lb}_m/\text{ft}^3$ were used for the reference temperature and density. The specific heat ratio γ , was set to 1.4. Since the experiment was incompressible, the reference Mach number M_r was set equal to 0.2 to minimize compressibility effects and, at the same time, achieve a reasonable convergence rate with the *Proteus* code. In the experiment, the Reynolds number based on the bulk velocity and the hydraulic diameter was 40,000. This value was therefore used as the reference Reynolds number Re_r in the calculation. The reference length L_r was set equal to 0.028658 ft. This value was computed from the definition of Re_r , where M_r and Sutherland's law were used to compute μ_r and μ_r , respectively.

4.3.2 Computational Coordinates. Figure 5 illustrates the computational grid for the S-duct, created using the GRIDGEN codes (Steinbrenner, Chawner, and Fouts, 1991). For clarity, the grid is shown only on three of the computational boundaries, and the points have been thinned by a factor of two in each direction. The boundary grids were first created using the GRIDGEN 2D program. The 3-D volumetric grid was then generated from the boundary grids using GRIDGEN 3D.

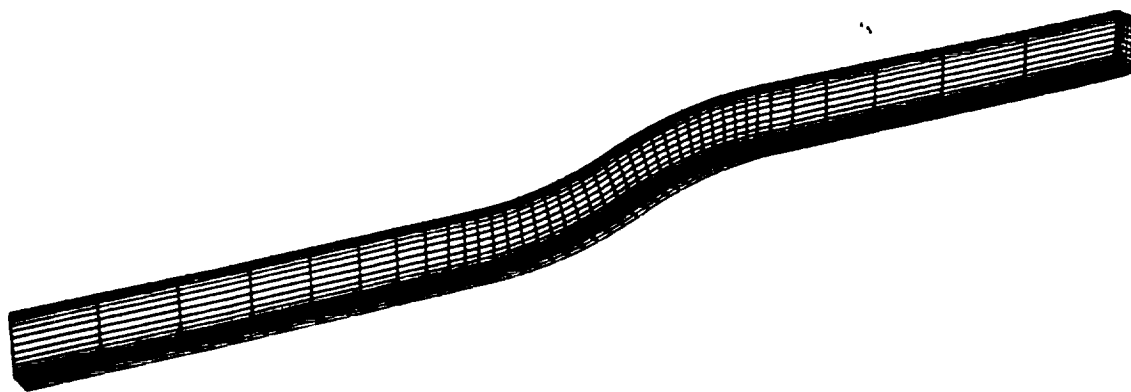


Figure 5. S-duct computational grid.

The computational grid extended from 7.5 hydraulic diameters upstream of the start of the first bend, to 7.5 hydraulic diameters downstream of the end of the second bend. The grid consisted of $81 \times 31 \times 61$ points in the ξ , η , and ζ directions, respectively. Since the S-duct is symmetric with respect to the $\eta = 1$ plane, only half of the duct was computed. To resolve the viscous layers, grid points were tightly packed near the solid walls using the default packing option in GRIDGEN 2D. At the grid point nearest the wall, the value of y^+ was about 0.5.

4.3.3 Initial Conditions. The computations were done in two separate major steps: a calculation using the Baldwin-Lomax turbulence model and a calculation using the Chien $k-\epsilon$ model. To start the Baldwin-Lomax calculations, the default initial profiles specified in subroutine INIT were used. Thus, the static pressure p was set equal to 1.0, and the velocity components u , v , and w were set equal to 0.0 everywhere in the duct. To start the Chien $k-\epsilon$ calculations, the initial values of u , v , w , p , and the turbulent viscosity μ_t were obtained from the Baldwin-Lomax solution. The initial values of k and ϵ were obtained using the default KEINIT subroutine in *Proteus*.

4.3.4 Boundary Conditions. For both calculations, constant stagnation enthalpy was assumed, eliminating the need for solving the energy equation. Therefore, only four boundary conditions were required for the mean flow at each computational boundary. In addition, for the Chien calculation, boundary conditions were required for k and ϵ at each computational boundary.

For the Baldwin-Lomax calculation, at the duct inlet the total pressure was specified as 1.02828, the gradient of u was set equal to zero, and the velocities v and w were set equal to zero. The inlet total pressure was calculated from the freestream static pressure and the reference Mach number using isentropic relations. At the duct exit, the static pressure was specified as 0.98416, and the gradients of u , v , and w were set equal to zero. The exit static pressure was found by trial and error in order to match the experimental mass flow rate. At the walls of the duct no-slip conditions were used for the velocities, and the normal pressure gradient was set to zero. Symmetry conditions were used in the symmetry plane.

For the Chien calculation, the boundary conditions for the mean flow were the same as for the Baldwin-Lomax calculation, with one exception. At the duct exit, the value of the static pressure was changed slightly, from 0.98416 to 0.98474, again in order to match the experimental mass flow rate. For the $k-\epsilon$ equations, at the upstream boundary the gradients of the turbulent kinetic energy k and the turbulent dissipation rate ϵ were set equal to zero for the first 20 time steps. After that time, the values of k and ϵ were kept constant. At the downstream boundary, the gradients of k and ϵ were set equal to zero. No-slip conditions were used at the solid boundaries, and symmetry conditions were used at the symmetry boundary.

4.3.5 Numerics. Both the Baldwin-Lomax and Chien calculations were run using a spatially varying time step. Since the flow field for the Baldwin-Lomax calculation was impulsively started from zero velocity everywhere, large CFL numbers specified at the very beginning of the calculation might result in an unphysical flow field and cause the calculation to blow up. Therefore, the calculations were run with a CFL number of 1 for the first 100 iterations, 5 for the next 200 iterations, and 10 for the remaining iterations. A total of 4,000 iterations was used for the Baldwin-Lomax calculation.

For the Chien case, a small CFL number was again used at the beginning of the calculation. The calculations were run with a CFL number of 1 for the first 120 iterations, 5 for the next 500 iterations, and 10 for the remaining iterations. A total of 2,520 iterations was used for the Chien calculation.

The constant coefficient artificial viscosity model was used for both cases, with $\epsilon_f = 2$ and $\epsilon_E^{(4)} = 1$.

The convergence criterion was that the average residual for each equation be less than 10^{-6} . However, both calculations were stopped before reaching this level of convergence when examination of several flow-related parameters indicated that the solution was no longer changing appreciably with time. The average residual at the end of the Baldwin-Lomax calculation ranged from 10^{-3} for the x -momentum equation to 3×10^{-5} for the continuity equation. For the Chien calculation the values were 3×10^{-4} for the x -momentum equation and 5×10^{-6} for the continuity equation. For both cases the residuals were continuing to drop when the calculations were stopped.

4.3.6 Computed Results. In Figure 6, the computed flow field from the Chien calculation is shown in the form of total pressure contours at five stations through the duct. (The upstream and downstream straight sections are not shown.) As the flow enters the first bend, the boundary layer at the bottom of the duct initially thickens due to the locally adverse pressure gradient in that region. In an S-duct, the high pressure at the outside (bottom) of the first bend drives the low energy boundary layer toward the inside (top) of the bend, while the core flow responds to centrifugal effects and moves toward the outside (bottom) of the bend. The result is a pair of counter-rotating secondary flow vortices in the upper half of the cross-section. These secondary flows cause a significant amount of flow distortion, as shown by the total pressure contours.

In the second bend, the direction of the cross-flow pressure gradients reverses, making the pressure higher in the upper half of the cross-section. However, the flow enters the second bend with a vortex pattern already established. The net effect is to tighten and concentrate the existing vortices near the top of the duct, in agreement with classical secondary flow theory. The resulting horseshoe-shaped distortion pattern at the exit of the second bend is typical of S-duct flows.

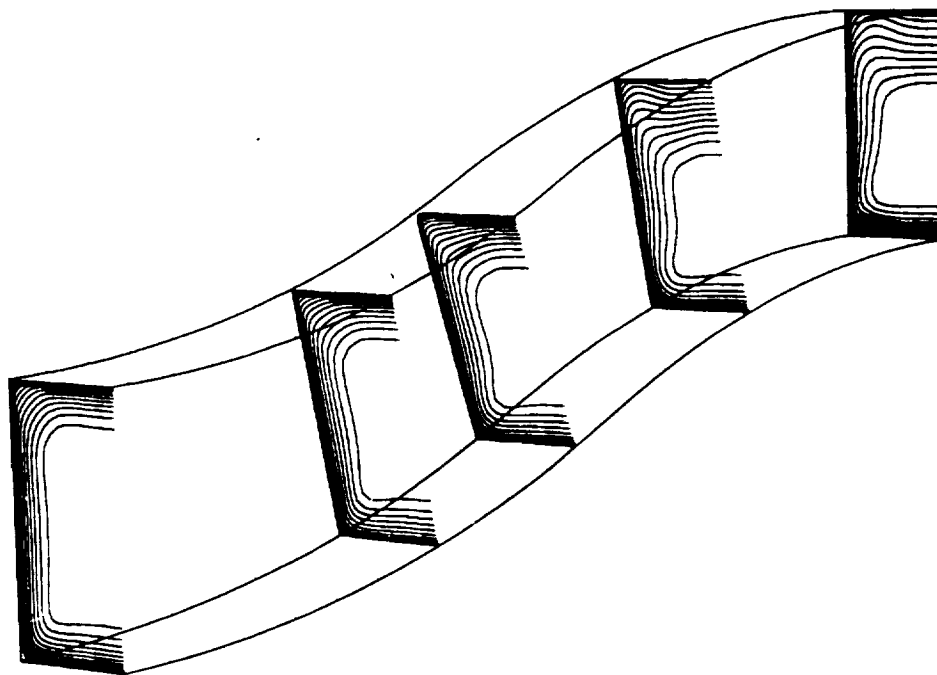


Figure 6. Computed total pressure contours for turbulent S-duct flow.

In Figure 7, the calculated wall pressure distribution is compared with the experimental data of Taylor, Whitelaw, and Yianneskis (1982). The agreement is very good. Both turbulence models correctly predicted the pressure trend and the pressure loss along the duct. The r and z coordinates noted in the legend are the same as those

defined by Taylor, Whitelaw, and Yianneskis.

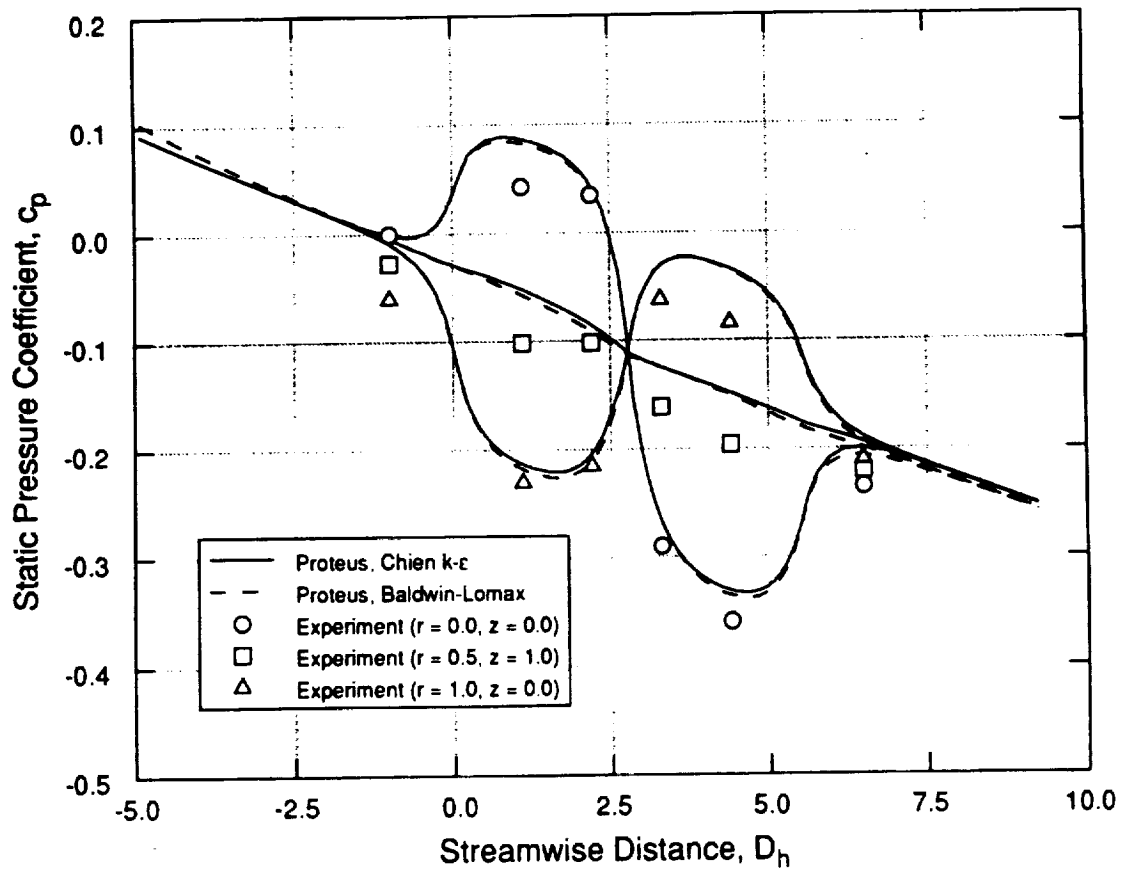


Figure 7. Computed surface static pressure distribution for turbulent S-duct flow.

In Figure 8, the experimental and computed velocity profiles in the symmetry plane are shown for the five streamwise stations that were surveyed in the experiment. These survey stations are at the same locations as the total pressure contours shown in Figure 6. The agreement between computation and experiment is excellent for both turbulence models. The asymmetry in the velocity profiles due to the pressure induced secondary motion is correctly predicted.

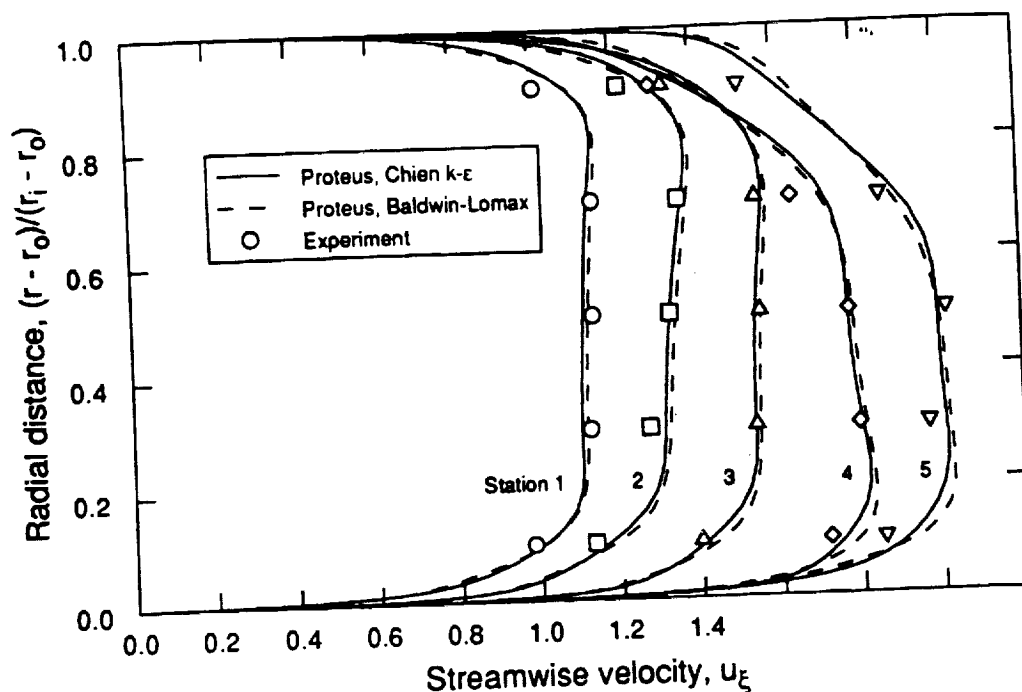


Figure 8. Computed streamwise velocity profiles for turbulent S-duct flow.

5. CONCLUDING REMARKS

The *Proteus* two- and three-dimensional Navier-Stokes codes recently developed at NASA Lewis have been described, and results have been presented from some of the validation cases. Version 1.0 of the two-dimensional code was released in late 1989 (Towne, Schwab, Benson, and Suresh, 1990), and version 2.0 was released in late 1991. Version 1.0 of the three-dimensional code was released in early 1992. Documentation for version 2.0 of the two-dimensional code and for version 1.0 of the three-dimensional code is available, but has not yet been formally published.

Current development work on the *Proteus* codes is being done to add a multiple-zone grid capability, a multi-grid convergence acceleration capability, and additional turbulence modeling options.

A wide variety of validation cases have been run, including: (1) several simplified flows for which exact Navier-Stokes solutions exist; (2) laminar and turbulent flat plate boundary layer flows; (3) two- and three-dimensional driven cavity flows; (4) flows with normal and oblique shock waves; (5) steady and unsteady flows past a cylinder; (6) developing laminar and turbulent flows in channels, pipes, and rectangular ducts; (7) steady and unsteady flows in a transonic diffuser; (8) flows in curved and S-shaped ducts; and (9) turbulent flow on a flat plate with a glancing shock wave. Current and future validation cases will emphasize three-dimensional duct flows and flows with heat transfer.

REFERENCES

- Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. (1984) *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, McGraw-Hill Book Company, New York.
- Baldwin, B. S., and Lomax, H. (1978) "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.
- Beam, R. M., and Warming, R. F. (1978) "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, No. 4, pp. 393-402.

- Bogar, T. J., Sajben, M., and Kroutil, J. C. (1983) "Characteristic Frequencies of Transonic Diffuser Flow Oscillations," *AIAA Journal*, Vol. 21, No. 9, pp. 1232-1240.
- Briley, W. R., and McDonald, H. (1977) "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics*, Vol. 24, pp. 373-397.
- Bui, T. T. (1992) "Implementation/Validation of a Low Reynolds Number Two-Equation Turbulence Model in the Proteus Navier-Stokes Code — Two-Dimensional/Axisymmetric," NASA TM 105619.
- Chien, K. Y. (1982) "Prediction of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," *AIAA Journal*, Vol. 20, No. 1, pp. 33-38.
- Conley, J. M., and Zeman, P. L. (1991) "Verification of the Proteus Two-Dimensional Navier-Stokes Code for Flat Plate and Pipe Flows," AIAA Paper 91-2013 (also NASA TM 105160).
- Grove, A. S., Shair, F. H., Petersen, E. E., and Acrivos, A. (1964) "An Experimental Investigation of the Steady Separated Flow Past a Circular Cylinder," *Journal of Fluid Mechanics*, Vol. 19, pp. 60-80.
- Hoffmann, K. A. (1989) *Computational Fluid Dynamics for Engineers*, Engineering Educational System, Austin, Texas.
- Hsieh, T., Bogar, T. J., and Coakley, T. J. (1987) "Numerical Simulation and Comparison with Experiment for Self-Excited Oscillations in a Diffuser Flow," *AIAA Journal*, Vol. 25, No. 7, pp. 936-943.
- Hsieh, T., Wardlaw, A. B., Jr., Collins, P., and Coakley, T. J. (1987) "Numerical Investigation of Unsteady Inlet Flow Fields," *AIAA Journal*, Vol. 25, No. 1, pp. 75-81.
- Hughes, W. F., and Gaylord, E. W. (1964) *Basic Equations of Engineering Science*, Schaum's Outline Series, McGraw-Hill Book Company, New York.
- Jameson, A., Schmidt, W., and Turkel, E. (1981) "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259.
- Pulliam, T. H. (1986) "Artificial Dissipation Models for the Euler Equations," *AIAA Journal*, Vol. 24, No. 12, pp. 1931-1940.
- Saunders, J. D., and Keith, Jr., T. G. (1991) "Results from Computational Analysis of a Mixed Compression Supersonic Inlet," AIAA Paper 91-2581.
- Steger, J. L. (1978) "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, No. 7, pp. 679-686.
- Steinbrenner, J. P., Chawner, J. P., and Fouts, C. L. (1991) "The GRIDGEN 3D Multiple Block Grid Generation System, Volume II: User's Manual," WRDC-TR-90-3022, Vol. II.
- Taylor, A. M. K. P., Whitelaw, J. H., and Yianneskis, M. (1982) "Developing Flow in S-Shaped Ducts," NASA CR 3550.
- Towne, C. E., Schwab, J. R., Benson, T. J., and Suresh, A. (1990) "PROTEUS Two-Dimensional Navier-Stokes Computer Code — Version 1.0, Volumes 1-3," NASA TM's 102551-3.

ANALYSIS OF FLUID/MECHANICAL SYSTEMS USING EASY5

Robert W. Clark, Jr.
McDonnell Douglas Space Systems Co.
Houston, Texas

Scott D. Arndt
McDonnell Douglas Space Systems Co.
Houston, Texas

Eric A. Hurlbert
National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas

SUMMARY

This paper illustrates how the use of a general analysis package can simplify modeling and analyzing fluid/mechanical systems. One such package is EASY5, a Boeing Computer Services product. The basic transmission line equations for modeling piped fluid systems are presented, as well as methods of incorporating these equations into the EASY5 environment. The paper describes how this analysis tool has been used to model several fluid subsystems of the Space Shuttle Orbiter.

INTRODUCTION

Modeling complex fluid/mechanical systems can involve difficulties beyond describing the system numerically. Not only does the task involve coding of the actual equations, the analyst is also faced with numerical integration of those equations, discretization of the system, and post-processing of the results. Thus, there exists a need for a tool which combines these processes into a single package. Boeing Computer Services EASY5 analysis program has been found to be one such tool which can be used to effectively model fluid/mechanical systems. With the advent of fast workstations based on RISC chips, graphically interfaced analysis programs for system analysis are highly efficient.

Modeling using EASY5 can be done in a finite-element type manner using modular subroutines. The user defines the behavior of a single element within the system (such as pipe flow or a spring-mass system) using the appropriate user-supplied equations and then discretizes the system as a combination of these elements, similar to other finite-element method routines. The features of this code benefit the user by providing nonlinear and linear analysis capability. Nonlinear time-domain simulations can be run using one of several different integration methods. This package also has the ability to linearize the system to provide transfer function, root locus, eigenvalue, as well as other types of analysis. Also contained within EASY5 is a plotting routine which can provide plots of results for the different types of analysis.

While any of the systems that could be modeled using EASY5 could also be modeled using FORTRAN, this type of software represents a convenient combination of many of

the tools which the analyst requires and significantly reduces time required to develop a new system simulation.

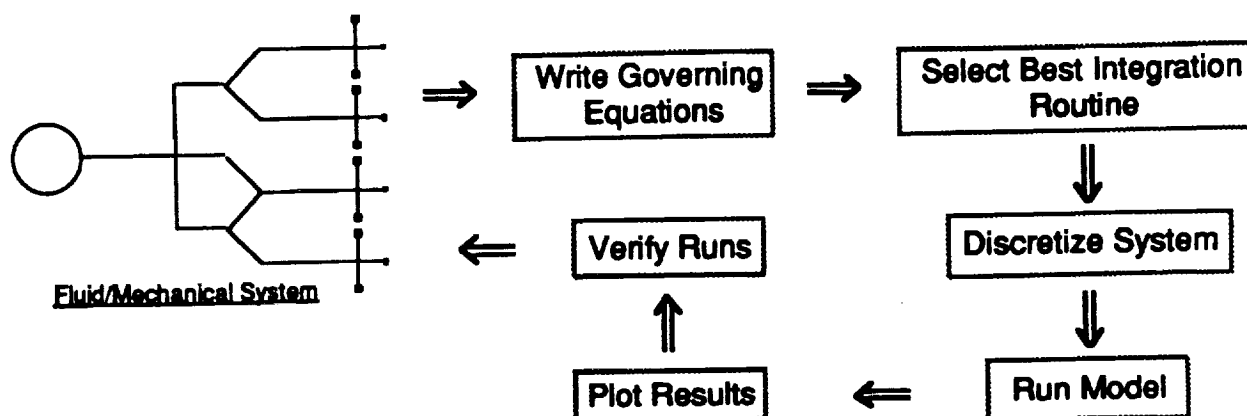


Figure 1. Flow chart of EASY5 modeling process.

THEORY

Most fluid/mechanical systems can be classified into subsets of similar components: pipe flow, pipe intersections (tees and crosses), orifices, volumes, and spring-mass systems. If the modeler has subroutines defining these components, they can be combined to represent complicated systems.

The basic building blocks for the fluid flow subroutines, or macros in EASY5 terminology, are the transmission line equations (ref. 1). The equations are listed below (see Figure 2 for notation).

$$I_i \ddot{m}_i = P_i - P_{i+1} - R_{fi} |\dot{m}_i| \dot{m}_i \quad (1)$$

$$C_i \dot{P}_i = \dot{m}_{i-1} - \dot{m}_i ; \quad i = 1, N \quad (2)$$

where:

I_i = inertance of the i^{th} fluid element,

C_i = capacitance of the i^{th} fluid element,

\dot{m}_i = mass flow into the $i+1$ element,

P_i = pressure at the center of the i^{th} fluid element

R_{fi} = resistance,

N = total number of fluid elements used to model a line segment

For a uniform line modeled with equal-length elements, the inertance, capacitance, flow resistance and temperature equations are the same for all elements and are given by (assuming one-dimensional flow and isentropic behavior):

$$I = \frac{L}{A} \quad (3)$$

$$C = \frac{V}{\gamma RT} \quad (4)$$

$$R_f = \frac{RT}{2A^2P} f \left(\frac{L + L_e}{D} \right) \quad (5)$$

$$T = T_0 \left(\frac{P}{P_0} \right)^{\frac{\gamma - 1}{\gamma}} \quad (6)$$

where:

- L = fluid element length
- A = flow area
- γ = polytropic process exponent
- T = temperature
- R = gas constant
- f = friction factor (pipe flow)
- D = line internal diameter
- L_e = equivalent length for minor losses

Equations 3, 4, and 5 specify the flow parameters for gas systems. These parameters can also be expressed for a liquid system by using the bulk modulus and density of the fluid.

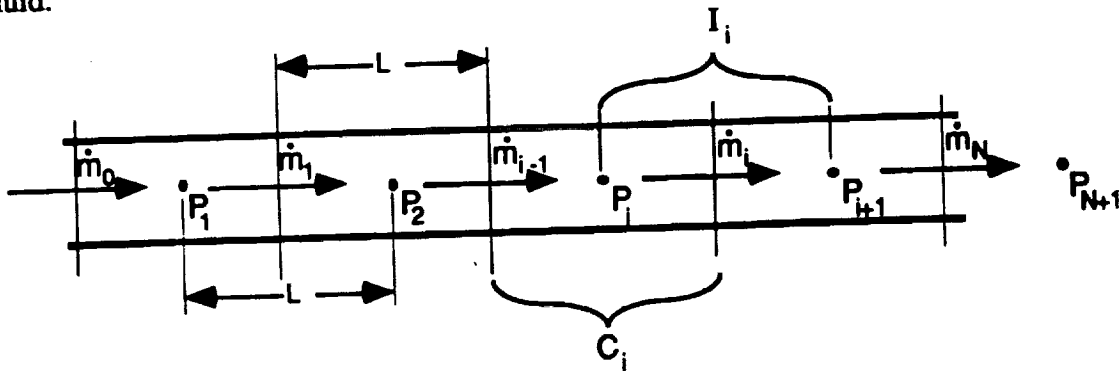


Figure 2. Typical discretization of a fluid line segment

The standard transmission line equations can be modified to handle flow through tees and crosses by using additional flow equations. Volumes of changing size can be modeled using Equation 7, which assumes an isentropic process.

$$\dot{P} = \gamma(\dot{m}RT - P\dot{V})/V \quad (7)$$

The mass flow rate through an orifice is given by the familiar relationship (ref. 2)

$$\dot{m} = C_D A \sqrt{\frac{2\gamma}{(\gamma-1)R} \frac{P_2}{\sqrt{T_2}}} \sqrt{\left(\frac{P_1}{P_2}\right)^{\frac{2}{\gamma}} - \left(\frac{P_1}{P_2}\right)^{\frac{\gamma+1}{\gamma}}} \quad (8)$$

where C_D = orifice discharge coefficient, A = orifice area, and the subscripts 1 and 2 denote pressure/temperature upstream and downstream, respectively. A valve can be modeled using a variable area orifice. To approximate the opening of the valve, it has been found that varying the area using a hyperbolic tangent function yields the best results. However, any type of continuous or discrete function could be used as long as the rates of change within the model do not become too large for the integration step size.

Because EASY5 requires systems of first order differential equations, spring-mass systems are modeled by breaking the system's second order differential equations into first order equations. For example, the governing differential equation for a spring-mass-damper system,

$$\ddot{x} = -(c\dot{x} + kx - F(t))/m \quad (9)$$

may be replaced by the following two first-order equations:

$$\dot{v} = -(cv + kx - F(t))/m \quad (10)$$

$$\dot{x} = v \quad (11)$$

APPLICATION

An EASY5 macro is very similar to a FORTRAN subroutine. The macro contains the code required to describe the behavior of a single model element, e.g., a transmission line element, spring-mass combination, etc. The parameters which define the physical characteristics of the element are inputs to the macro, as are the boundary conditions for that element as calculated by an adjacent element. The outputs of the macro are the values calculated using the code within the macro and the specified inputs. A model is then built by linking a series of macros together using their inputs and outputs.

For example, consider the three element section of a model shown in Figure 3. An acoustic line is being modeled using a macro named 'TR' (EASY5 macro names consist of 2 characters). The acoustic line macro is a combination of the pressure/flow differential equations, isentropic temperature relationship, and a curve fit of the Moody diagram. The macro first calculates the current temperature assuming an isentropic process. Next, the macro uses a logic block to determine which way flow is moving. After the flow direction is determined, the friction factor is calculated using the Reynold's Number and the equations describing the Moody diagram. The flow and pressure derivatives are then calculated and integrated. These outputs are then used as inputs to other elements.

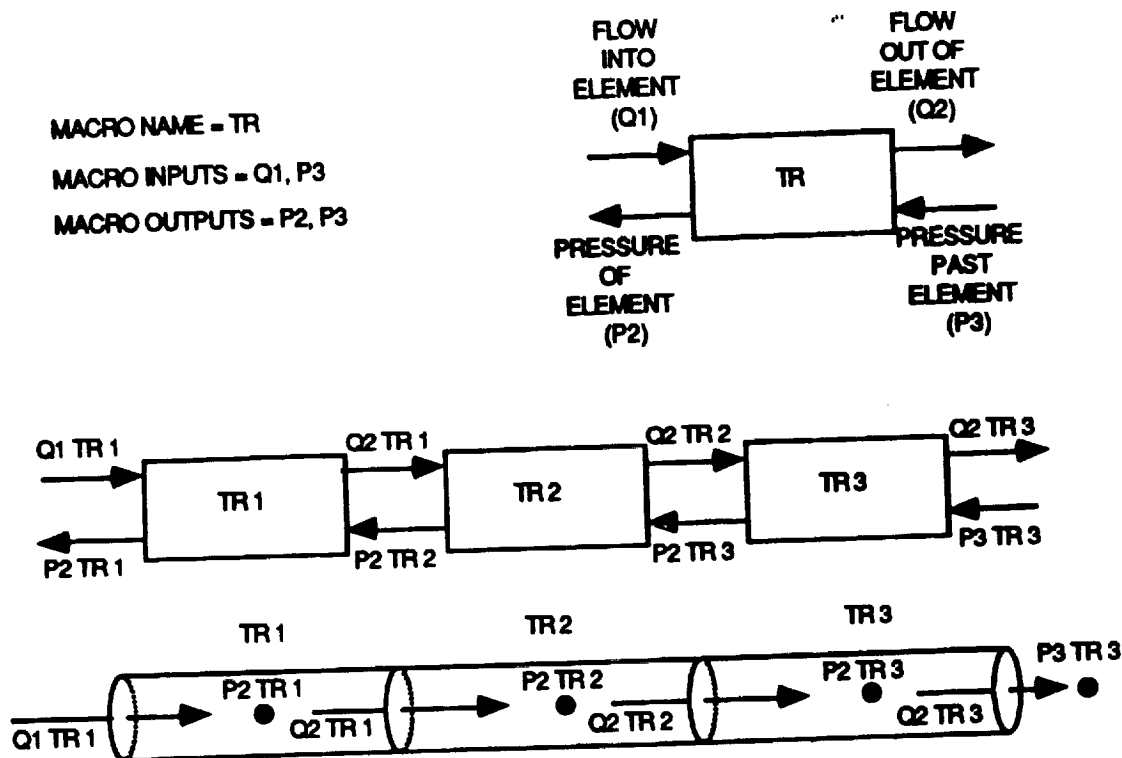


Figure 3. Acoustic Line Connection and a Simple Three Element Model

When modeling an acoustic line, the line is broken up into a series of elements. The length of each element is based on the highest frequency of interest and the length of the line. The usual FEM rules regarding the minimum and maximum number of elements in a line apply to this type of modeling. An acoustic line macro is used for each element. The pressure derivative is based on the flow out of the previous element and the flow out of the current element. The flow derivative is based on the pressure of the previous element and the pressure of the next line element. Therefore, the flow of the previous element and the pressure of the next line element must be inputs to the current line element.

A long length of line can require an excessive number of transmission line elements. In order to minimize the effort required to build the model, a multiple element transmission line macro was developed. The code internal to the macro is set up in an array format. The user specifies the number of sub-elements to be contained within the element, ranging from 1 to 999. This development greatly reduces the amount of time required to develop a model of a system.

The time step used for nonlinear time-domain simulations varies depending on the nature of the model. The optimum time step is found through an iterative process for fixed time step integrators, while variable time step integration schemes have logic for adjusting integration time step to maintain solution accuracy with the largest acceptable time step. Too large of a time step results in numerical error due to large rates of change. Too small of a time step can cause excess round-off error. The optimum time step for fixed step solutions has been found to be one which, when reduced, gives results identical to those of the previous step size. The recurrence formula for the wave equations must be considered when choosing a time step size. Therefore, the following relationship needs to be considered,

$$\frac{1}{c} \leq \frac{\Delta t}{\Delta x} \quad (12)$$

where Δt is the time step size, Δx is the element length and c is the speed of sound of the media being modeled. A detailed explanation can be found in reference 3.

EASY5 offers several different types of integration algorithms. These include:

Fixed-Step

Euler (1st order)
Huen (2nd order)
Fixed-Step Runge-Kutta (4th order)

Variable Step

BCS Gear
Adams-Moulton
Stiff Gear
Variable-Step Runge-Kutta

The variable step integration schemes adjust the integration step size based on how fast the system states are changing. Ideally, these methods would be desirable for use since they represent a potential execution time savings. However, it has been the authors' experience that the variable step methods are not particularly compatible with the macros that have been developed to model fluid systems, due to the quadratic damping term and the large pressure derivatives associated with small elements. Typically, the integrator ends up iterating excessively trying to optimize the step-size, thereby greatly increasing the execution time. Good results have been obtained using the variable-step methods on spring-mass systems.

Another nonlinear analysis feature of EASY5 is steady state analysis. The steady state command returns the equilibrium operating condition of the model. The model rates of change are essentially zero for this analysis.

EASY5 is also capable of linear dynamics analysis. This is done by linearizing the state equations in the model by perturbing them about the operating point to create a linear perturbation model. This linear model can then be used for other types of analysis such as transfer function, root locus, closed loop eigenvalue and other frequency domain analyses.

EXAMPLE 1 - 750 PSIA MPS HELIUM SUPPLY REGULATOR

Background

In this example the authors were asked to investigate a problem with the Space Shuttle main propulsion system (MPS) 750-psia helium pressure regulator. Two regulators experienced full-open failures due to high frequency (900 Hz), high amplitude oscillations. The failures took place on a new test stand which was constructed to replace the original regulator qualification stand after it was destroyed in the collapse of the building in which it was located.

The authors were tasked to develop dynamic models of the test stand as well as models of all three MPS engine helium supply system configurations utilizing an existing model of the regulator developed by the vendor. The purpose of the models was to determine the source of the oscillations, evaluate potential for oscillations on the Orbiter, and to test possible solutions for correcting the problem.

Modeling Effort

The EASY5 software was selected for this modeling effort. It was not possible to directly convert the vendor's regulator model into EASY5. Therefore, the model had to be created with EASY5 macros using the existing model as a guide. EASY5 macros of the components discussed in the Theory section of this paper were assembled to represent the actual regulator (see Figure 4 for regulator schematic). The model consisted of one spring-mass macro (containing 21 degrees of freedom), twelve flow (tube, annular and orifice) macros, and nine volume macros. The spring-mass macro contained the necessary equations to model the movement of the poppet, valves and metal bellows. The hard stops in the regulator were modeled by using bi-linear springs.

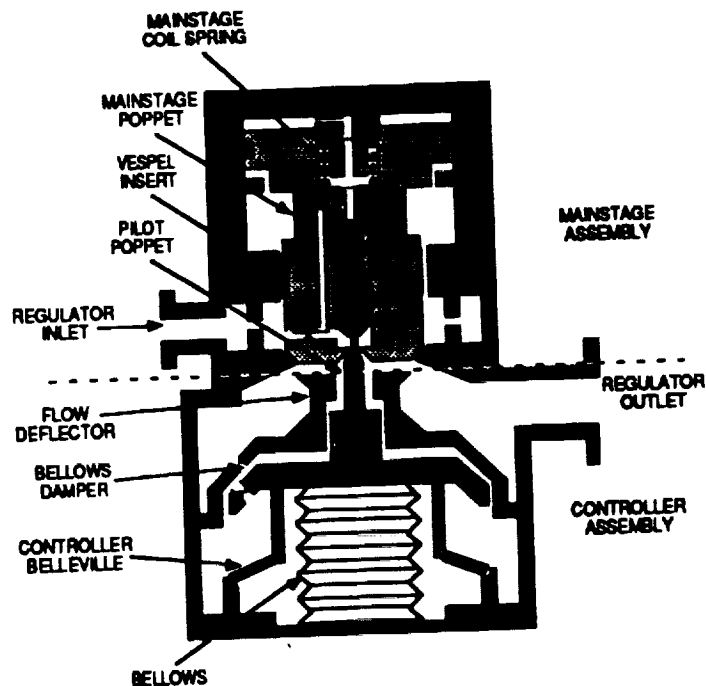


Figure 4. MPS helium supply regulator schematic

Analysis

Using the transfer function option of EASY5, it was determined that there was a 180 degree phase shift between the pressure sensed and actual pressure of the regulator's exit at the frequency range that the oscillations occurred. The shift would cause the regulator to reinforce any pressure oscillations occurring downstream of the regulator in this frequency range.

A model of the complete newly constructed verification test stand was developed. The oscillatory behavior of the regulator was duplicated using the time simulation option and it matched the first acoustic mode downstream of the regulator. The new test stand line configuration's fundamental frequency coincidentally matched that of the regulator's bellows, which lead to fatigue failure of the bellows. Models of the complete Orbiter MPS helium supply system were also constructed (Figure 5). Each engine supply system consisted of approximately 1000 degrees-of-freedom.

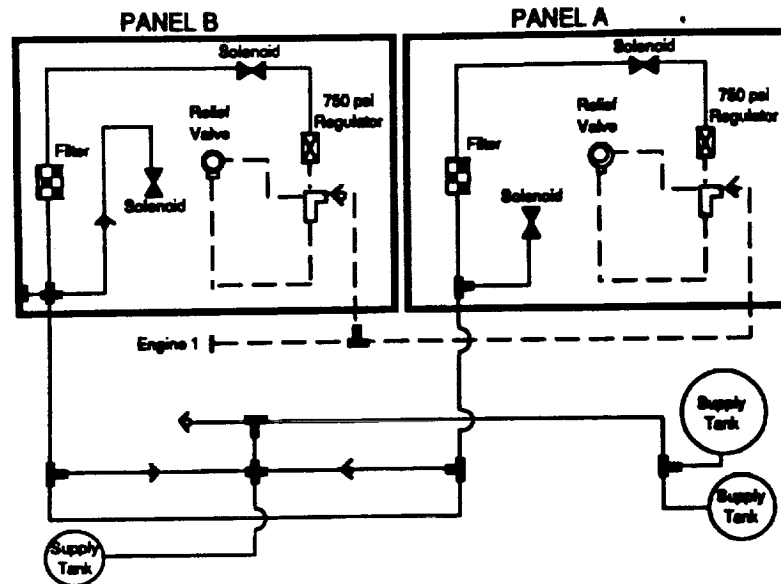


Figure 5. Diagram of Engine 1 helium supply system

The EASY5 model of the regulator was used to guide and evaluate design changes proposed by the vendor. The final design showed stable operation in both tests of actual hardware and in numerical time simulations with the math model. Figures 6 and 7 show a Bode plot and simulation results of the regulator before and after the redesign.

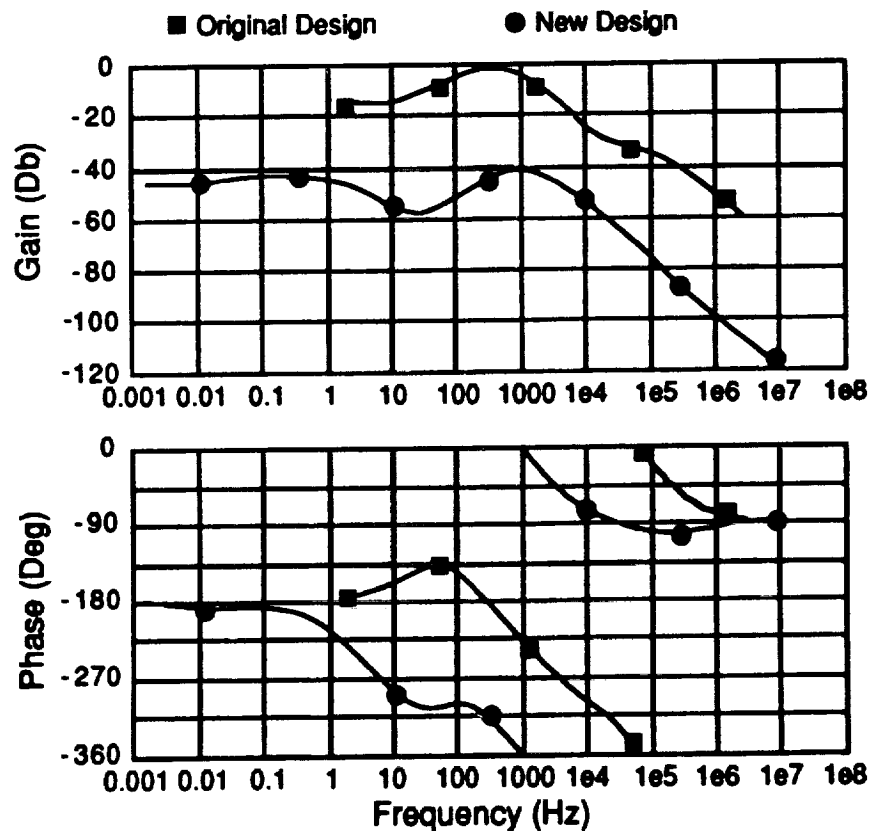
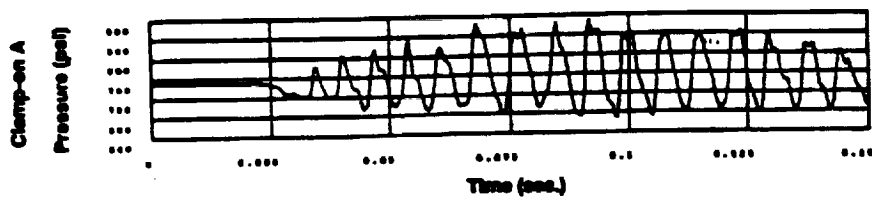


Figure 6. Bode plot of regulator outlet pressure to controller pressure transfer function.



Regulator Pressure Prior to Redesign



Regulator Pressure After Redesign

Figure 7. Time simulations of the regulator before and after redesign.

EXAMPLE 2 - PRCS THRUSTER

This second example describes the use of the EASY5 software in developing propellant feedline dynamic models of the Orbiter's Primary Reaction Control System (PRCS) thruster test stands. Over the course of the modeling project, the models evolved from a simple waterhammer analysis to a complex two-phase flow analysis of the chug stability of the thruster.

Background

Combustion stability testing of the PRCS thruster involves injecting helium into the propellant feedlines in order to provide a combustion disturbance. The injection rate is not a precisely known quantity. The test stand is designed to produce a nominal flow of helium during steady state conditions. However, due to ignition and shutdown transients, the flow of helium into the thruster can vary widely over time. For this reason an analytical model was desired to predict the amount of helium ingested by the thruster. The propellants for the thruster are monomethylhydrazine and nitrogen tetroxide, both of which are liquids at the operating pressure and temperature.

Modeling

The test stand models were developed using macros similar to those used for the MPS helium regulator project. The models are comprised of single- and two-phase elements. Line elements several inches upstream of the helium injection point are capable of two-phase flow representation, while the remainder of the transmission line elements are single-

phase (see Figure 8). The two-phase macros assume a homogeneous gas-liquid mixture, ideal and isentropic behavior of the gas phase, and are based on equation (13):

$$\dot{P} = \frac{\gamma R T \dot{m}_g + \frac{\gamma \dot{m}_l P}{\rho_l}}{V_g + \frac{\gamma V_l P}{B}} \quad (13)$$

where

B = liquid bulk modulus

V_g = gas volume of element

\dot{m}_g = mass flow of gas

\dot{m}_l = mass flow of liquid

ρ_l = density of liquid

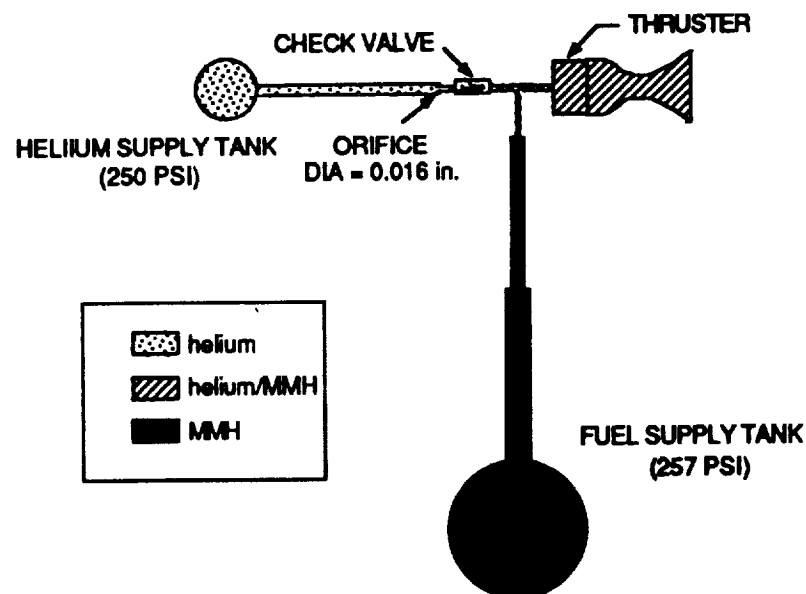


Figure 8. Schematic of PRCs Test Stand Model

Initial simulations used a time history table of thruster chamber pressure measured in test firings as the boundary condition at the end of the propellant feedline. Based on the steady state flow rate and pressure drop, the resistance of the line could be fine-tuned to achieve the required flow parameters. Time-domain simulations used fourth-order Runge-Kutta as the integration method, with an integration step size of 1.0E-05 seconds.

Results from these models were in the form of time history plots of system parameters. Sample results are shown in Figures 9 and 10. Figure 9 illustrates the behavior of fuel feedline pressure just upstream of the thruster. Information from these simulations was also used to size the test stand feedline lengths to obtain a waterhammer frequency similar to the actual vehicle installation. Figure 10 shows the helium ingestion time history for a single 160 millisecond firing. Simulations can be executed sequentially such that the ending conditions for one simulation are the initial conditions for the next. For the PRCS modeling task, this allowed a sequence of firings to be simulated so that the amount of helium residing in the lines would build as the test firing sequence progressed.

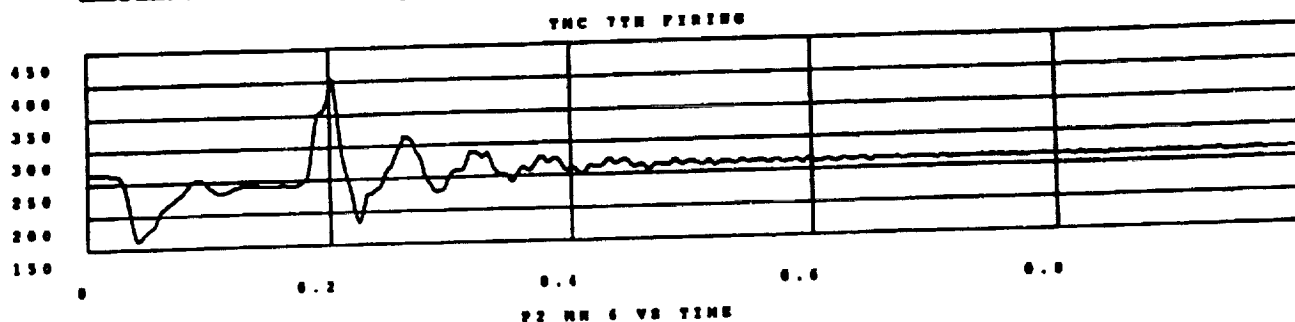


Figure 9. Typical simulation result.

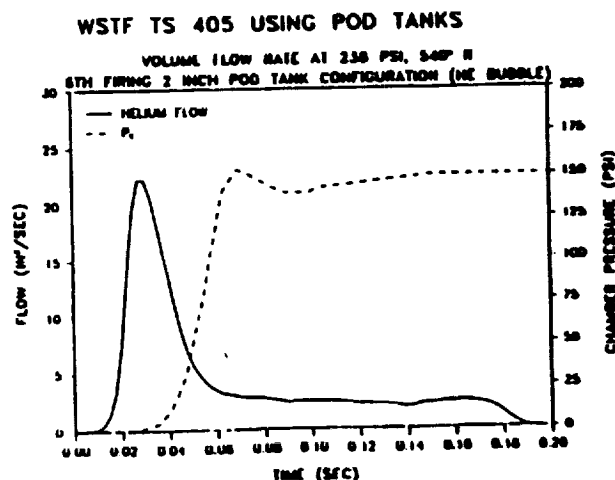


Figure 10. Helium injection profile.

Evolution of the PRCS Models

The PRCS modeling effort expanded beyond the scope of the initial test stand models. In order to understand the mechanism behind the low frequency (600-1000 Hz) chug mode of the thruster, a more detailed model of the thruster valves, manifolds, and injectors was developed. The valve model is similar in concept to the fluid/mechanical model developed for the MPS regulator task. Variable area orifices were used to represent the opening and closing of valve passages as the valve poppet moved. The stiffness of the poppet spring is represented by tabular data taken from tests conducted during the valve development program. Leak rates around the poppet seals are simulated by not allowing the variable area orifices to close completely.

A diagram of the model schematic is shown in Figure 11. Test stand vibration, which may contribute to some of the high amplitude pressure and acceleration oscillations

observed during tests, is included in the model. The test stand is treated as a single degree-of-freedom system having mass, stiffness, and damping characteristics close to that of the stand. Test stand motion is applied to the fluid system through macros which have moving boundaries. The test stand velocity is applied to these elements as an element wall velocity, which drives the element pressure derivative. Also included in the model is the combustion timelag. The flow out of the last injector element is delayed from combusting (expanding into gas) by a specified amount of time. This is accomplished through an EASY5 continuous delay macro. The chamber pressure is calculated based on the capacitance of the chamber, the amount of fuel and oxidizer flowing into the chamber, and the amount of gas flowing out of the chamber. The amount of gas flowing out of the chamber is determined using the characteristic velocity (c^*).

Due to the small size of the injector, very small elements were necessary to obtain the required fidelity. The size of these elements dictated that the integration step size also be small. The optimum step size was found to be $1\text{E-}07$ seconds, using fourth-order Runge-Kutta as the integration method.

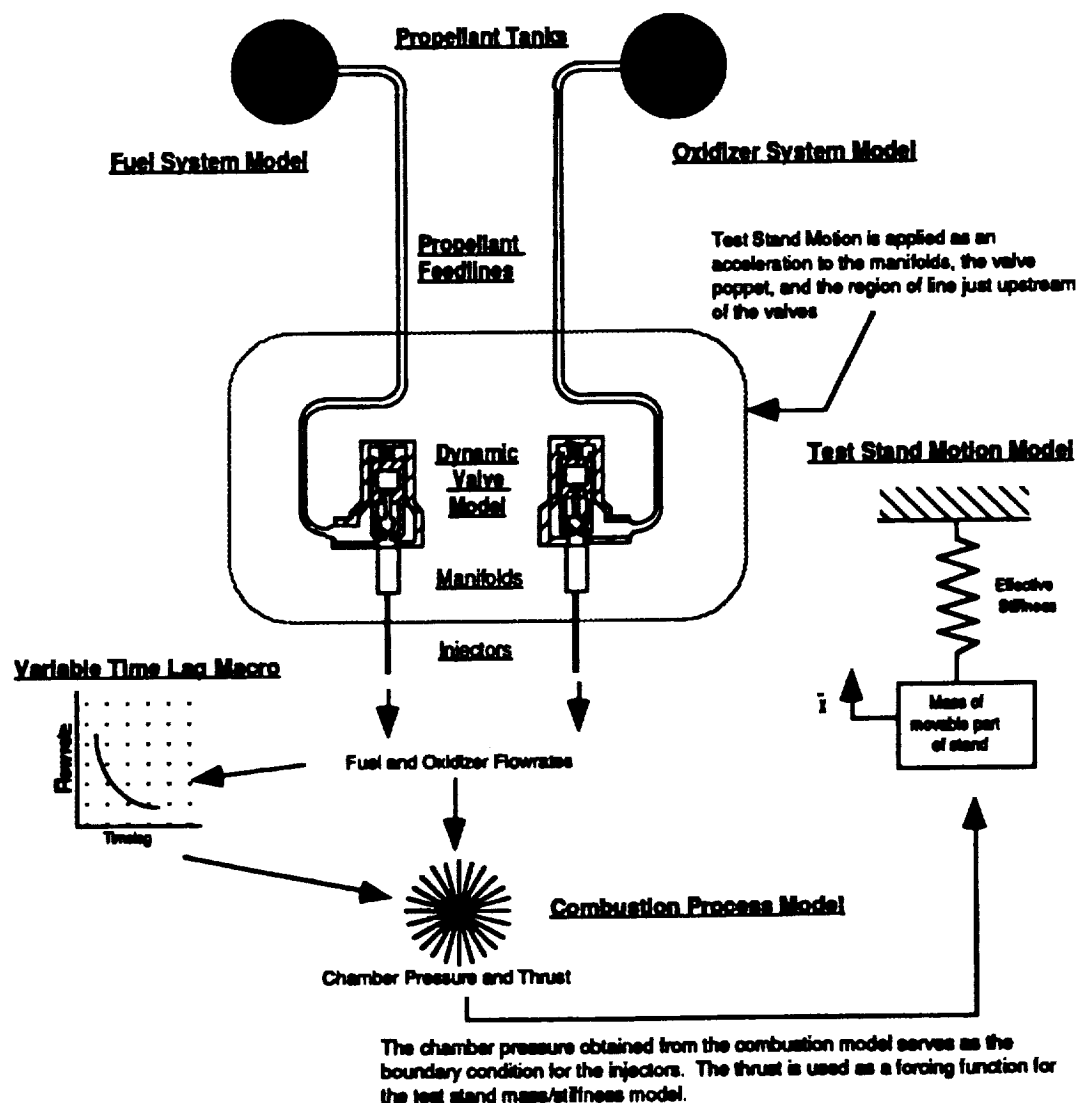


Figure 11. Schematic of detailed PRCS thruster model.

CONCLUSIONS

The use of a general analysis package for fluid/mechanical system modeling has been demonstrated. The general transmission line equations and numerical approximations of other fluid system components have been successfully integrated into the EASY5 analysis program. The combination of integration routines, graphics capability, and pre- and post-processor has proven to be effective and convenient for modeling these types of systems. High fidelity models of several complex non-linear fluid, structural, and mechanical system interactions were developed which correlated well with test data and provided a basis for analyzing and eliminating causes of adverse dynamic interactions.

NASA-JSC Propulsion Branch is continuing to use EASY5 for other propulsion systems. A substantial set of macros and models have been developed which allow quick and accurate analytical results to be obtained for a wide variety of propulsion fluid and mechanical systems.

REFERENCES

1. Brown, F.T.: The Transient Response of Fluid Lines. *Journal of Basic Engineering*, Dec. 1962, pp. 547-553.
2. Andersen, B.W.: *The Analysis and Design of Pneumatic Systems*. John Wiley & Sons, Inc., New York, 1967.
3. Crandall, S.H.: *Engineering Analysis*. McGraw-Hill Inc., New York, 1956.

ACOUSTIC MODES IN FLUID NETWORKS

C.D. Michalopoulos
McDonnell Douglas Space Systems Co.
Houston, Texas

Robert W. Clark, Jr.
McDonnell Douglas Space Systems Co.
Houston, Texas

Harold H. Doiron
McDonnell Douglas Space Systems Co.
Houston, Texas

SUMMARY

Pressure and flow rate eigenvalue problems for one-dimensional flow of a fluid in a network of pipes are derived from the familiar transmission line equations. These equations are linearized by assuming small velocity and pressure oscillations about mean flow conditions. It is shown that the flow rate eigenvalues are the same as the pressure eigenvalues and the relationship between line pressure modes and flow rate modes is established. A volume at the end of each branch is employed which allows any combination of boundary conditions, from open to closed, to be used.

The Jacobi iterative method is used to compute undamped natural frequencies and associated pressure/flow modes. Several numerical examples are presented which include acoustic modes for the Helium Supply System of the Space Shuttle Orbiter Main Propulsion System.

It should be noted that the method presented herein can be applied to any one-dimensional acoustic system involving an arbitrary number of branches.

INTRODUCTION

Often in the analysis of dynamic responses of piped fluid networks, a preliminary "quick look" at acoustic mode shapes and frequencies of the system is a useful diagnostic tool prior to the initiation of more detailed diagnostic testing or modeling efforts. Knowledge of the fundamental and higher order response mode frequencies of the system based on linear analysis allows for rapid assessment of modes which may couple dynamically with devices such as regulators and check valves. This provides valuable diagnostic information when troubleshooting dynamic problems with these types of devices. Knowledge of pressure and flow mode shapes can provide guidance on positioning of high frequency pressure and flow transducers during testing. Such information can be used to infer magnitude of pressure and flow oscillations in regions of the fluid system where measurements cannot be made due to various practical limitations typically encountered on operational systems.

During the course of diagnostic studies of several dynamic phenomena with regulators, check valves, propellant feed systems and rocket engines of the Space Shuttle Orbiter

spacecraft over the last four years, the authors developed a systematic fluid element approach to the analysis of related piped fluid networks. These modeling procedures were incorporated into a FORTRAN computer program called ACLMODES.

This paper presents derivations of basic building block equations used in the program, illustrates numerical accuracy of the computer code on several problems with known closed-form solutions and illustrates how the program was used to analyze several dynamic phenomena associated with piped fluid networks of the Orbiter spacecraft.

GOVERNING EQUATIONS

The basic equations employed in this paper are the familiar pneumatic/hydraulic transmission line equations which govern one-dimensional transient flow. These equations for an unbranched acoustic line are first re-cast in matrix form. Then, this matrix formulation is generalized for systems of branched acoustic lines, referred to as "fluid networks".

Matrix Form of Equations for Unbranched Acoustic Lines

The ordinary differential equations governing one-dimensional flow of an ideal gas, in terms of volumetric flow, are (see Figure 1 for notation)

$$I_i \dot{Q}_i = P_i - P_{i+1} - R_{fi} |Q_i| Q_i \quad (1)$$

$$C_i \dot{P}_i = Q_{i-1} - Q_i ; \quad i = 1, N \quad (2)$$

where:

- I_i = inertance of the i^{th} fluid element,
- C_i = capacitance of the i^{th} fluid element,
- Q_i = mass flow into the $i+1$ element,
- P_i = pressure at the center of the i^{th} fluid element
- R_{fi} = resistance,
- N = total number of fluid elements used to model a line segment

For a uniform line modeled with equal-length elements, the inertance, capacitance, and flow resistance are the same for all elements and are given by:

$$I = \frac{\rho L}{A} \quad (3)$$

$$C = \frac{AL}{\gamma p R T} \quad (4)$$

$$R_f = \frac{\rho}{2A^2} f \left(\frac{L + L_e}{D} \right) \quad (5)$$

where:

- L = fluid element length
- A = flow area
- γ = polytropic process exponent
- T = temperature
- R = gas constant
- ρ = density
- f = friction factor (pipe flow)
- D = line internal diameter
- L_e = equivalent length for minor losses

It should be noted that Eq. (1) may be easily derived by integrating once the one-dimensional momentum equation and neglecting the convective terms. Equation (2) is the equation of conservation of mass for isentropic flow of an ideal gas. These equations are derived in References [1], [2], and [3].

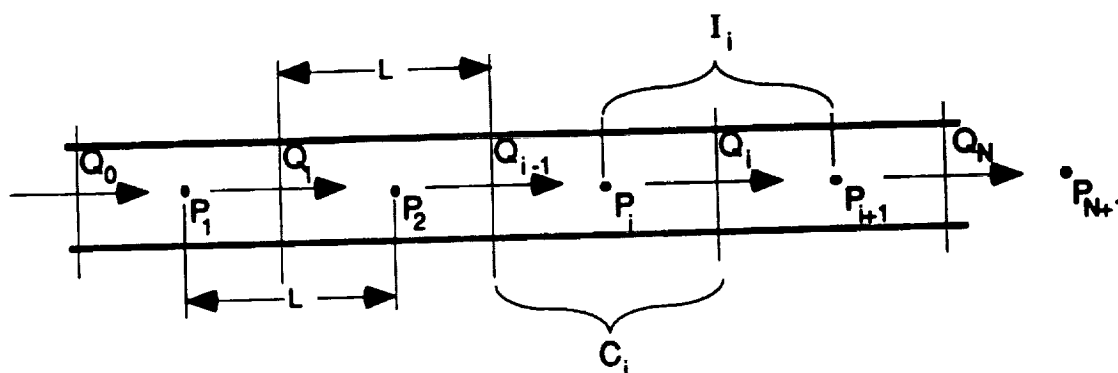


Figure 1. Typical discretization of a line segment.

The sets of Eqs. (1) and (2) may be written in matrix form in the special case of $R_f \equiv 0$, that is, for the undamped system. The matrix equations are

$$A \dot{Q} + B P = F_1 \quad (6)$$

$$C \dot{P} - B^T Q = F_2 \quad (7)$$

where the superscript denotes the transpose of the matrix, and:

$$A = \begin{bmatrix} I_1 & & & \\ & I_2 & & \\ & & \ddots & \\ & & & I_N \end{bmatrix} \quad (8)$$

$$\mathbf{C} = \begin{bmatrix} C_1 & & & \bigcirc \\ & C_2 & & \\ & & \ddots & \\ \bigcirc & & & C_N \end{bmatrix} \quad (9)$$

$$\mathbf{B} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 \end{bmatrix} \quad (10)$$

$$(B_{ij} = -1, \quad B_{i+1} = 1, \quad \text{otherwise } B_{ij} = 0)$$

$$\mathbf{Q} = \begin{Bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{Bmatrix}; \quad \mathbf{P} = \begin{Bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{Bmatrix} \quad (11)$$

$$\mathbf{F}_2 = \begin{Bmatrix} Q_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix}; \quad \mathbf{F}_1 = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -P_{N+1} \end{Bmatrix} \quad (12)$$

Differential Equations for Acoustic Lines with Branches

Consider a branched acoustic system such as that shown in Figure 2. The end volumes V_1 , V_2 and V_3 are used in the formulation for generalizing the boundary conditions. It should be noted (see Appendix A) that $V = 0$ represents a closed end while $V = \infty$ is an open end.

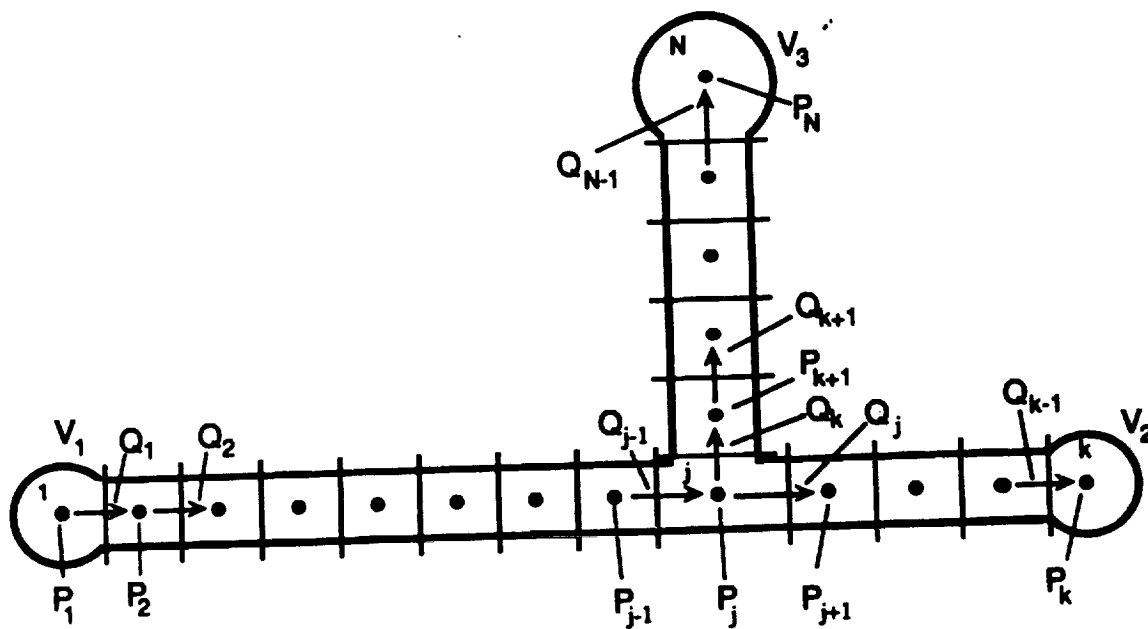


Figure 2. Example of an acoustic line with a branch.

For simplicity, damping is neglected in this section. Equations (6) and (7) are applicable in this case, but the matrix B has a different structure from that of Eq. (10). Note that Eqs. (1) and (2) with $R_{fi} = 0$ apply at all elements with some modifications at the ends (elements 1, K , and N) and element j , where the branch connects to the main line. The first-order equations for these special elements are:

$$C_1 \dot{P}_1 = Q_0 - Q_1 \quad (13)$$

$$C_j \dot{P}_j = Q_{j-1} - Q_j - Q_k \quad (14)$$

$$C_k \dot{P}_k = Q_{k-1} \quad (15)$$

$$C_N \dot{P}_N = Q_{N-1} \quad (16)$$

$$I_k \dot{Q}_k = P_j - P_{k+1} \quad (17)$$

with $C_1 = \frac{V_1}{\gamma RT}$. The B matrix in this case (one branch) has the following structure:

$$\begin{aligned} B_{ii} &= -1; \quad i = 1, N; \quad i \neq k \\ B_{kj} &= -1 \\ B_{i, i+1} &= 1; \quad i = 1, N \end{aligned} \quad (18)$$

Note that B is an $N \times (N + 1)$ rectangular matrix.

An example on the structure of B:

$$\begin{aligned} j &= 2 \\ k &= 4 \\ N &= 7 \end{aligned}$$

$$\begin{aligned} &(7 \times 8) \\ \mathbf{B} &= \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{aligned} \quad (19)$$

The extension to acoustic lines with multiple branches, or fluid networks, is straight forward. Obviously, the structure of B depends on the numbering system used.

Linearized Form of Governing Equations

For small pressure/flow oscillations, it can be shown that

$$\mathbf{A} \ddot{\mathbf{q}} + \mathbf{D} \dot{\mathbf{q}} + \mathbf{E}_Q \mathbf{q} = \mathbf{g}_1 \quad (20)$$

$$\mathbf{C} \ddot{\mathbf{p}} + \mathbf{H} \dot{\mathbf{p}} + \mathbf{E}_p \mathbf{p} = \mathbf{g}_2 \quad (21)$$

where A and C are given by Eqs. (8) and (9) respectively. The diagonal damping matrix D is defined by

$$\mathbf{D} = \begin{bmatrix} \beta_1 & & & \\ & \beta_2 & & \\ & & \ddots & \\ & & & \beta_N \end{bmatrix}$$

where $\beta_i = 2R_i Q_i^0$ is the linear damping coefficient, Q_i^0 being the mean (steady) flow rate.

Note that the vectors \mathbf{q} , \mathbf{p} , \mathbf{f}_1 and \mathbf{f}_2 are defined according to Eqs. (11) and (12), with Q, P, F_1 and F_2 replaced by q , p , f_1 and f_2 respectively. The matrices H, \mathbf{E}_q , \mathbf{E}_p , \mathbf{g}_1 and \mathbf{g}_2

in Eqs. (20) and (21) are given by

$$\mathbf{H} = \mathbf{B}^T \mathbf{A}^{-1} \mathbf{D} \mathbf{B}^T{}^{-1} \mathbf{C} \quad (22)$$

$$\mathbf{E}_Q = \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^T \quad (23)$$

$$\mathbf{E}_P = \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \quad (24)$$

$$\mathbf{g}_1 = \dot{\mathbf{f}}_1 - \mathbf{B} \mathbf{C}^{-1} \mathbf{f}_2 \quad (25)$$

$$\mathbf{g}_2 = \dot{\mathbf{f}}_2 + \mathbf{B}^T \mathbf{A}^{-1} (\mathbf{f}_1 + \mathbf{D} \mathbf{B}^T{}^{-1} \mathbf{f}_2) \quad (26)$$

ACOUSTIC MODES IN FLUID NETWORKS

The undamped natural frequencies and mode shapes for a fluid network are determined from Eqs. (20) and (21) with $\mathbf{D} = 0$. Setting the right hand of these equations equal to zero, the free, undamped flow/pressure oscillations in a fluid network are governed by

$$\mathbf{A} \ddot{\mathbf{q}} + \mathbf{E}_Q \dot{\mathbf{q}} = 0 \quad (27)$$

$$\mathbf{C} \ddot{\mathbf{p}} + \mathbf{E}_P \dot{\mathbf{p}} = 0 \quad (28)$$

where \mathbf{A} , \mathbf{C} , \mathbf{E}_Q and \mathbf{E}_P are defined by Eqs. (8), (9), (23) and (24) respectively.

The eigenvalue problem associated with Eq. (27) is

$$\mathbf{A} \mathbf{X} = \lambda_Q \mathbf{E}_Q \mathbf{X} \quad (29)$$

where $\lambda_Q = \frac{1}{\omega_Q^2}$ and \mathbf{X} is a flow eigenvector or flow mode.

The eigenvalue problem associated with Eq. (28) is

$$\mathbf{C} \mathbf{Y} = \lambda_P \mathbf{E}_P \mathbf{Y} \quad (30)$$

where $\lambda_P = \frac{1}{\omega_P^2}$ and \mathbf{Y} is a pressure eigenvector or pressure mode.

It can be easily shown that

$$\omega_Q = \omega_P \quad (31)$$

and

$$\mathbf{Y} = \mathbf{C}^{-1} \mathbf{B}^T \mathbf{X} \quad (32)$$

Computer Program ACLMODES

A computer program, referred to as ACLMODES, was developed which computes the natural frequencies and associated flow rate/pressure modes for an acoustic line network. The program can accommodate any number of branches with any combination of boundary conditions (ranging from closed to open at each end). The input to the program is relatively simple due to its capability of generating acoustic elements with identical properties.

The program employs the Jacobi Iterative Method to solve the eigenvalue problem defined by either Eq. (29) or Eq. (30). Line pressure modes are computed either directly from Eq. (30) or using Eq. (32).

NUMERICAL RESULTS AND DISCUSSION

Three examples demonstrating capabilities of the ACLMODES program are shown in the following paragraphs. The first example is a comparison of a simple system's frequencies predicted by ACLMODES and the closed-form solution shown in Appendix A. The other examples are actual applications of ACLMODES on Space Shuttle fluid line systems.

Numerical Test Case

Example 1 is a test case consisting of a 100 inch long pipe of 0.5 inch I.D. filled with helium and a volume on both ends. Three different combinations of end volumes, shown in Table 1, were used. Note that all volumes are in cubic inches. Case A represents an open-closed boundary conditions and Case B a closed-closed.

	VOLUME A	VOLUME B
CASE A	0.00001	10000
CASE B	0.00001	0.00001
CASE C	0.1	10

Table 1. Volume sizes for ACLMODES test case.

ACLMODES was used to determine the first three natural frequencies of each of the three cases. Each case was repeated with four different element lengths to evaluate solution accuracy versus number of line elements employed. The closed-form solution shown in Appendix A was then used to calculate the frequencies of the three cases. The ACLMODES results are shown in Table 2 together with the closed-form solution.

	MODE	NUMBER OF ELEMENTS				CLOSED
		10	20	50	100	FORM
CASE A	1	100.30	100.38	100.40	100.40	100.40
	2	298.23	300.31	300.98	300.98	301.00
	3	488.85	498.43	501.13	501.52	501.64
CASE B	1	199.83	200.45	200.62	200.64	200.65
	2	394.73	399.66	401.04	401.24	401.30
	3	579.92	596.40	601.06	601.73	601.95
CASE C	1	145.02	145.09	145.10	145.11	145.11
	2	320.47	322.36	322.89	322.96	322.98
	3	502.28	511.27	513.81	514.17	514.28

Table 2. Acoustic Frequencies (Hz) of Cases A, B and C.

The percent difference between the values ACLMODES predicted and that of the closed-form solution are shown in Table 3.

	MODE	NUMBER OF ELEMENTS			
		10	20	50	100
CASE A	1	0.10	0.02	0.00	0.00
	2	0.92	0.23	0.01	0.01
	3	2.55	0.64	0.10	0.02
CASE B	1	0.41	0.10	0.01	0.00
	2	1.64	0.41	0.06	0.01
	3	3.66	0.92	0.15	0.04
CASE C	1	0.06	0.01	0.01	0.00
	2	0.78	0.19	0.03	0.01
	3	2.33	0.59	0.09	0.02

Table 3. Percent Error of Cases A, B and C Compared to Closed-Form Solution

These numerical results show excellent agreement with the closed-form solution results. As expected, there is improvement in accuracy of the numerical solution as the number of line elements increases. A general rule of thumb for an acceptable line element length required to obtain accurate numerical results is $\omega L/c < 0.5$, where ω is the estimated circular frequency of the mode sought in rad/sec, L is the line element length (inch) and c is the speed of sound in the fluid (inch/sec). Figure 1 shows percent error versus $\omega L/c$ for all values in Table 3.

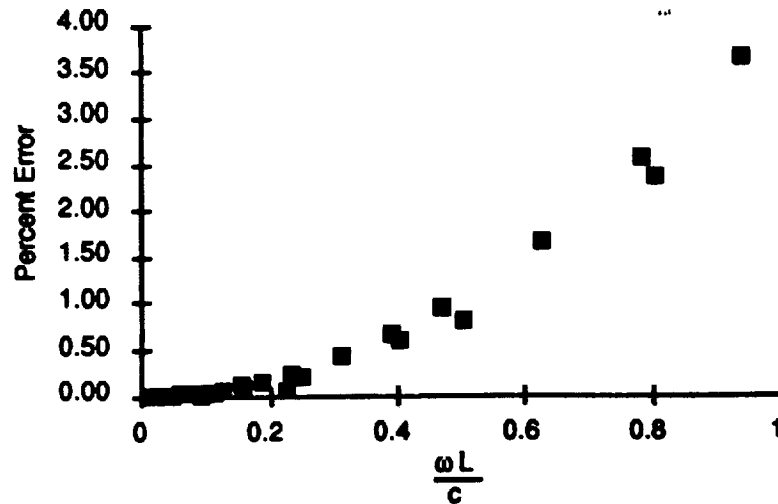


Figure 1. Percent error versus $\omega L/c$

It can be seen that when the condition $\omega L/c < 0.5$ is not met the percent error is greater than 1.

Test Stand Line Dynamics

Stability testing of the Primary Reaction Control System (PRCS) thruster at the NASA White Sands Test Facility (WSTF) required that the test stand have similar line dynamics to that of the Space Shuttle Orbiter. This is because the PRCS thruster is a pressure-fed engine so the pressure recovery or waterhammer of the line governs the start-up transients. Therefore, a simple line having similar waterhammer characteristics to that of the aft PRCS feed system, which is a fairly complicated system with many branches and twelve primary thrusters (see Figure 2), was desired.

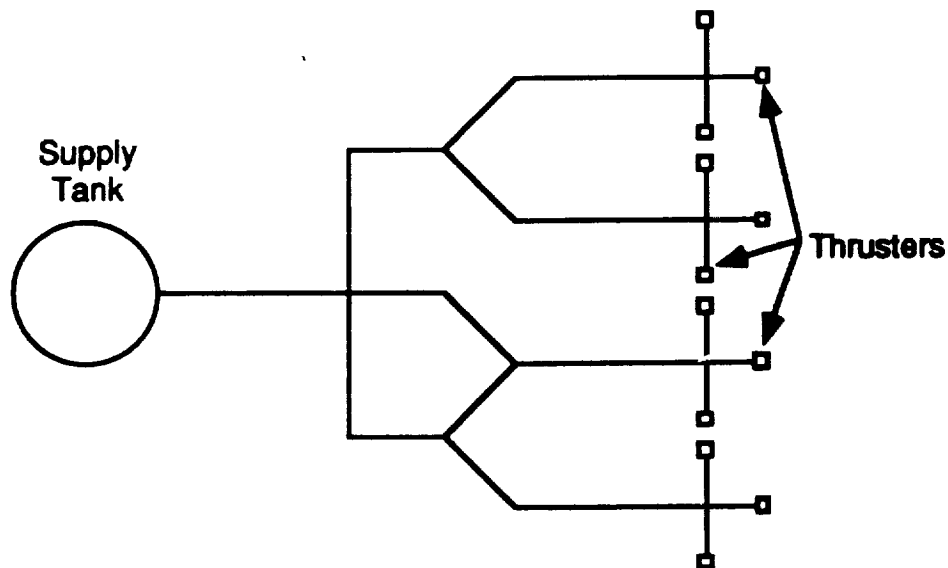


Figure 2. Schematic of the Space Shuttle's PRCS aft fuel supply system.

The original idea was to use similar line diameters to that of the vehicle and the average distance from the supply tank to the thrusters as the test stand line length. However, it was not obvious that this would yield the same dynamics as the vehicle so a model of each configuration was constructed.

The first mode of the proposed test stand fuel line was found using ACLMODES and is shown in Figure 3. As expected it appears to be an open-closed mode with a frequency of 65 Hz. The first mode of the vehicle's piping system was found to have a frequency of 40 Hz and is shown in Figure 4. The difference in frequency was not acceptable so the test stand line was reconfigured to have the same first natural frequency as the vehicle feed system.

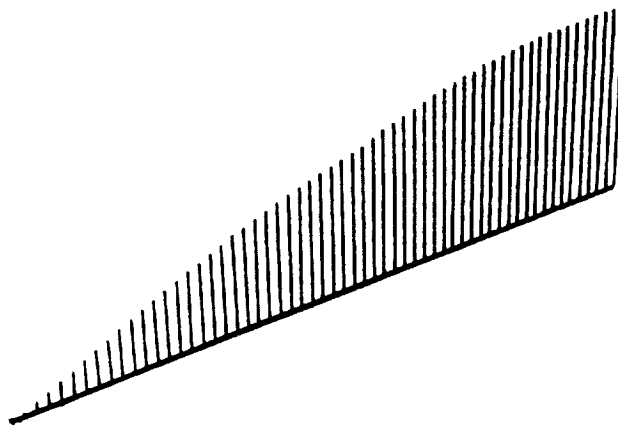


Figure 3. First pressure mode of simple feed system.

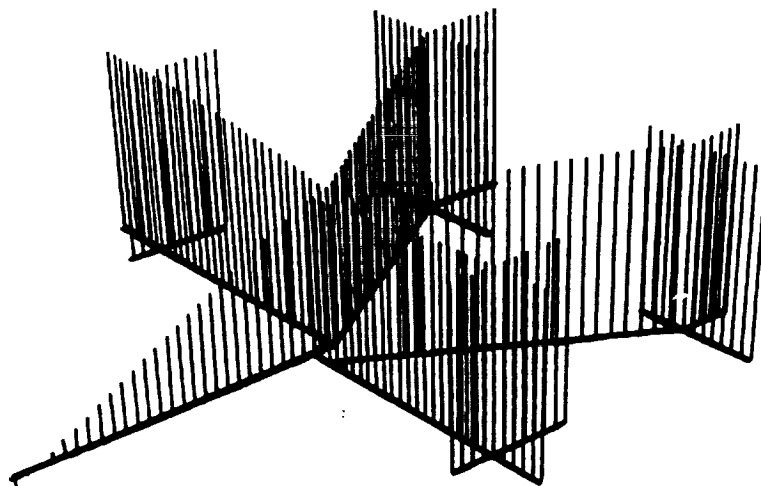


Figure 4. First pressure mode of the Space Shuttle's PRCS aft fuel supply system.

Space Shuttle Main Propulsion System Helium Supply System

The regulators in the helium supply system for Space Shuttle Main Propulsion System (MPS) were experiencing oscillations. These oscillations were seen both on test stands as well as on the vehicle. It was believed that the source of the oscillations was the regulators coupling with the downstream line acoustics.

Each engine has its own helium supply system. A helium supply system consists of high pressure supply tanks, tubing leading up to two panels in parallel, and lines from the panels rejoining and continuing on to the engine. A panel consist of a regulator and relief valve as well as several solenoids and check valves. Only the lines downstream of the regulators were of interest, so they were all that was modeled (see Figure 5).

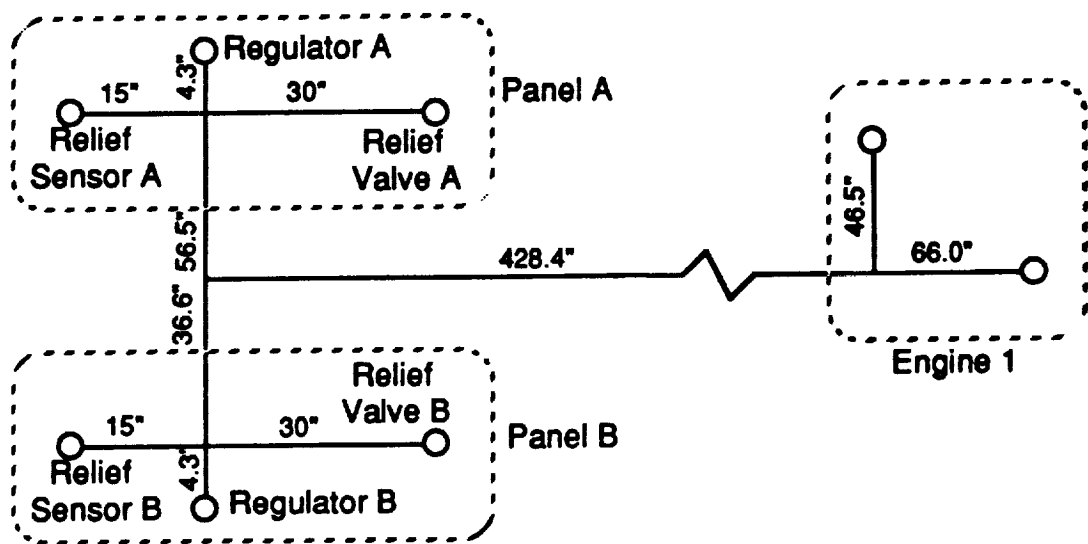


Figure 5. MPS Engine 1 helium schematic of lines downstream of regulators.

Figure 6 through 8 show the first, fourth and eighteen pressure modes predicted by ACLMODES for the Engine 1 helium supply system. The fourth mode is shown again in Figure 9 as a flow mode.

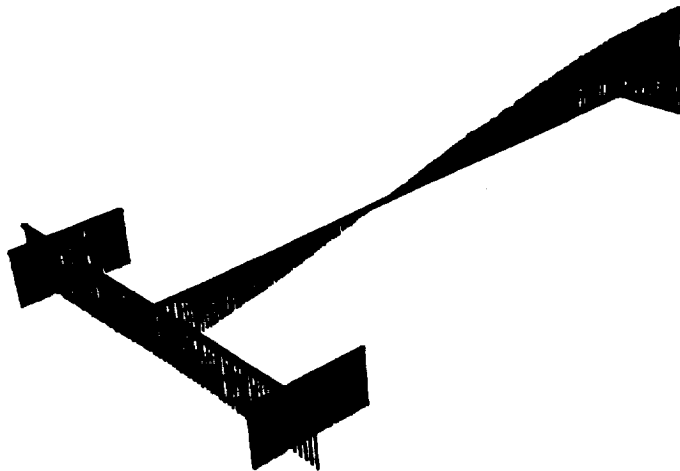


Figure 6. First pressure mode of the Engine 1 helium lines downstream of regulators.

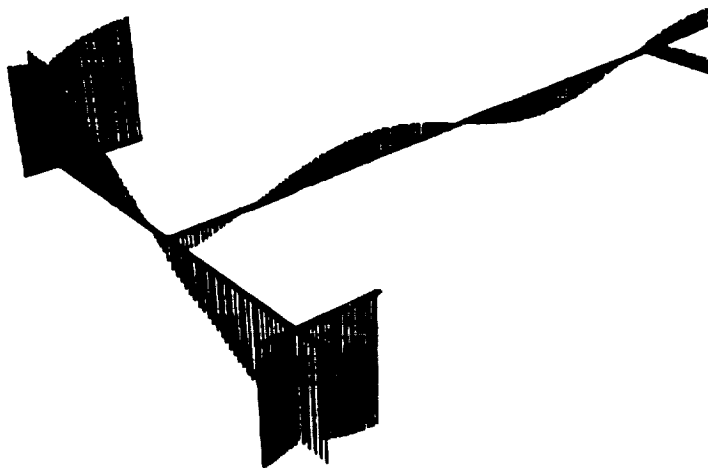


Figure 7. Fourth pressure mode of the Engine 1 helium lines downstream of regulators.

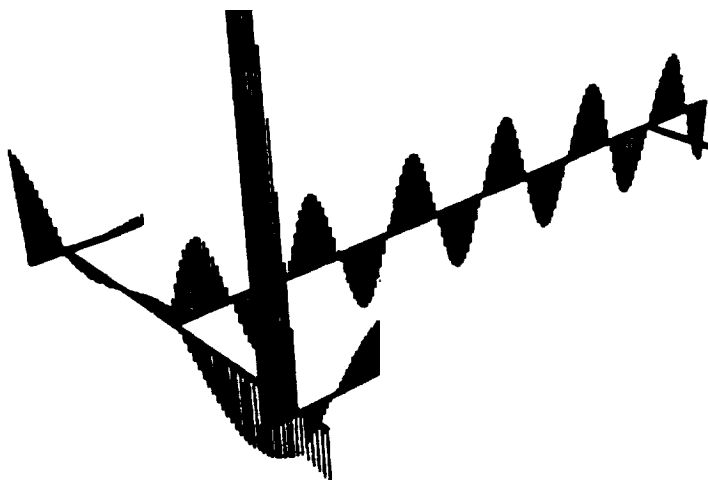


Figure 8. Eighteenth pressure mode of the Engine 1 helium lines downstream of regulators.

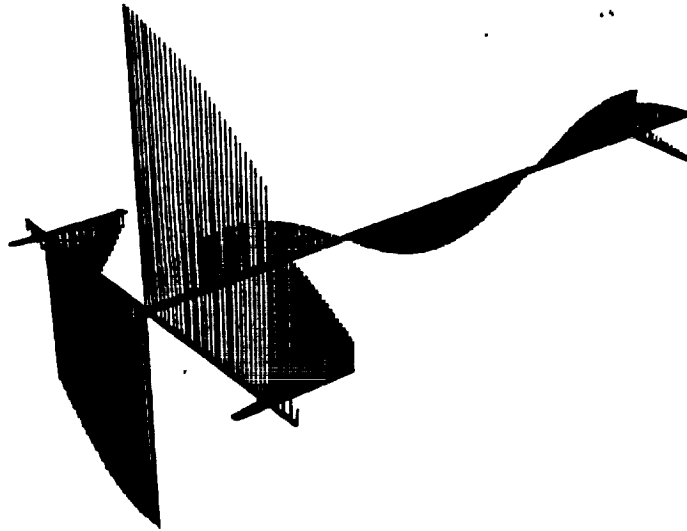


Figure 9. Fourth flow mode of the Engine 1 helium lines downstream of regulators.

Not all regulators oscillated on the vehicle and those that did, oscillated at different frequencies depending on flow demand and number of regulators in use. One mode of oscillation was around 115 to 120 hertz, with the regulators oscillating out of phase with each other. This mode was predicted by ACLMODES and can be seen in Figure 7 and Figure 9.

The pressure mode shapes were also used to determine if the pressure oscillations being measured by a transducer were representative of the oscillations at the regulators. This was done by examining the modes with frequencies near the frequency of interest and determining if the pressure amplitude at the transducer was being attenuated or amplified compared to that of the regulator.

CONCLUSIONS

The method presented herein has proven to be a very useful and accurate tool for determining dynamic characteristics of complex fluid networks, such as pressure recovery and oscillatory behavior. When implemented in a computer code and coupled with a plotting routine, this technique can graphically show vital information about the behavior of a fluid system impossible to obtain with hand calculations.

REFERENCES

1. Schuder, C.B. and R.C. Binder, "The Response of Pneumatic Transmission Lines to Step Inputs," *Journal of Basic Engineering, Trans. ASME*, pp. 578-584, December 1959.
2. Schwirian, R.E., "Multidimensional Waterhammer Analysis Using a Node-Flow Link Approach," *FED Vol. 30*, pp. 69-77, 1985.
3. Schwirian, R.E., et. al., "A Method for Predicting Pump-Induced Acoustic Pressures in Fluid-Handling Systems," *ASME PVP Vol. 63*, pp. 167-184, 1982.

APPENDIX A

Modes of a Straight Acoustic Line with General End Conditions

The classical wave equation for a straight tube in terms of volumetric flow rate is

$$c^2 \frac{\partial^2 Q}{\partial x^2} = \frac{\partial^2 Q}{\partial t^2} \quad (\text{A.1})$$

The boundary conditions for the system shown below are derived from the continuity equation and the definition of fluid capacitance.

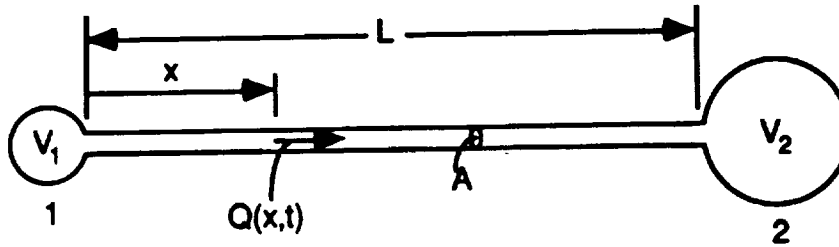


Figure A-1. Straight acoustic line.

The continuity equation is

$$\rho c^2 \frac{\partial Q}{\partial x} = -A \frac{\partial P}{\partial t} \quad (\text{A.2})$$

where

ρ = fluid density
 c = acoustic velocity
 Q = volumetric flow rate
 P = pressure
 A = flow area

But

$$\frac{\partial P}{\partial t} = \frac{1}{C} (Q_{in} - Q_{out}) \quad (\text{A.3})$$

where C is the fluid capacitance which is given by

Gas:
$$C = \frac{V}{\gamma P} = \frac{V}{\gamma RT} = \frac{V}{\rho c_s^2}$$

Liquid:
$$C = \frac{V}{B} = \frac{V}{\rho c_L^2} \quad (B = \text{Bulk Modulus})$$

Substitution of (A.3) in (A.2) yields

$$\rho c^2 \frac{\partial Q}{\partial x} = -A \frac{1}{C} (Q_{in} - Q_{out})$$

or,

$$\frac{\partial Q}{\partial x} = -\frac{A}{V} (Q_{in} - Q_{out}) \quad (A.4)$$

For end 1, $x = 0$,

$$Q_{in} = 0, \quad Q_{out} \equiv Q|_{x=0}$$

Thus,

$$\frac{\partial Q}{\partial x} \Big|_{x=0} = \frac{A}{V} Q|_{x=0} \quad (A.5)$$

Similarly, for end 2, $y=L$,

$$Q_{in} \equiv Q|_{x=L}, \quad Q_{out} = 0$$

from which

$$\frac{\partial Q}{\partial x} \Big|_{x=L} = -\frac{A}{V} Q|_{x=L} \quad (A.6)$$

The general solution of Eq. (A.1) is

$$Q(x,t) = T(t) (D_1 \sin \frac{\omega}{c} x + D_2 \cos \frac{\omega}{c} x) \quad (A.7)$$

where

$$T(t) = B_1 \sin \omega t + B_2 \cos \omega t \quad (A.8)$$

Substitution of conditions (A.5) and (A.6) into Eq. (A.7) leads to the following frequency equation

$$\Omega \tan \Omega = \frac{1 + \frac{\alpha_1}{\alpha_2}}{\alpha_1 - \frac{1}{\alpha_2 \Omega^2}} \quad (A.9)$$

where $\Omega = \frac{\omega}{c} L$ is a nondimensional frequency, $\alpha_1 = \frac{V_1}{AL}$ and $\alpha_2 = \frac{V_2}{AL}$

Special Cases

1. $V_2 \rightarrow \infty$ (open end)

$$\Omega \tan \Omega = \frac{1}{\alpha_1}$$

2. $V_2 \rightarrow 0$ (closed end)

$$\tan \Omega = -\alpha_1 \Omega$$

THE PROGRAM FANS-3D (FINITE ANALYTIC NUMERICAL SIMULATION 3-DIMENSIONAL) AND ITS APPLICATIONS

Ramiro H. Bravo¹
Tri-State University
Angola, Indiana 46703

Ching-Jen Chen²
University of Iowa
Iowa City, Iowa 52242

SUMMARY

In this study, the program named FANS-3D (Finite Analytic Numerical Simulation - 3 Dimensional) is presented. FANS-3D was designed to solve problems of incompressible fluid flows and combined modes of heat transfer. It solves problems with conduction and convection modes of heat transfer in laminar flow, with provisions for radiation and turbulent flows. It can solve singular or conjugate modes of heat transfer. It also solves problems in natural convection, using the Boussinesq approximation. FANS-3D was designed to solve heat transfer problems inside one, two and three dimensional geometries that can be represented by orthogonal planes in a Cartesian coordinate system. It can solve internal and external flows using appropriate boundary conditions such as symmetric, periodic and user specified.

INTRODUCTION

The program FANS-3D solves one, two and three dimensional fluid flow and heat transfer problems that involve conduction and convection modes of heat transfer in incompressible laminar flow, with provisions for radiative heat transfer and buoyant and turbulent flows. It also solves problems in natural convection using the Boussinesq approximation. Using this feature, the program may also solve mixed natural and forced convection problems. Furthermore, it can solve individual modes of heat transfer as well. The program FANS-3D solves any geometry that can be represented by orthogonal planes in a Cartesian coordinate system. The program can solve internal and external flows using appropriate boundary conditions such as symmetric, periodic and user specified. The program is designed to have the same performance in all directions, in this way, any problem can be solved in the most convenient orientation.

The program FANS-3D is based on the 19-point Finite Analytic Method. It uses the SIMPLEC iterative method suggested by Van Doormal and Raithby (1984) to solve the coupled Navier-Stokes equations. The discretization of variables is done following a new staggered grid layout. The resulting system of algebraic equations is solved by different methods, including ADI, SSOR and Conjugate Gradient.

The program has two modules; a graphics and a computational. The graphics module, named GRAPH3D, was written in FORTRAN 77 for Apollo workstations. This program displays the geometry of solution and the results in the three-dimensional space. These are

¹ Assistant Professor Department of Aerospace and Mechanical Engineering.

² Professor and Chairman Department of Mechanical Engineering.

presented in the form of velocity vectors, profiles, and color contours with shading of any variable. These graphic results can also be sent to monochrome and color printers.

The computational module of the program is divided in two parts. The first, which is accessible to the user, should be modified according to the problem. The second part is fixed and does not require user intervention. These two parts must be bound together to create the computational 'run' file. Both parts of the computational module are written in standard FORTRAN 77 language. In this form the program can be easily ported to almost any machine.

THE STRUCTURE OF THE PROGRAM FANS-3D

As mentioned above, the program FANS-3D is based on the 19 Point Finite Analytic Method. The basic idea of this method is illustrated here. Details of its derivation are given by Bravo et. al. (1991). This method is derive for the general transport equation

$$\phi_{xx} + \phi_{yy} + \phi_{zz} = D\phi_i + 2A\phi_x + 2B\phi_y + 2C\phi_z - Sh \quad (1)$$

using the analytic solutions of simplified forms of it. To start, this equation is first locally linearized in the three-dimensional element shown in figure 1. To this effect, the coefficients A, B, C, D and the source term S are assumed constants and equal to their values at the center of the element, i.e.

$$\phi_{xx} + \phi_{yy} + \phi_{zz} = D_p\phi_i + 2A_p\phi_x + 2B_p\phi_y + 2C_p\phi_z - S_p \quad (2)$$

The p subindices mean that these coefficients are considered constant inside the element and equal to their values at the center 'p'. For example, if ϕ is the u velocity component in a laminar flow and R the Reynolds number, then $A_p = Ru_p/2$, $B_p = Rv_p/2$, $C_p = Rw_p/2$, $D = R$ and $S_p = -RP_x$. Equation (2) is then solved in the planes $x=0$, $y=0$ and $z=0$ and shown in figure 2. These two-dimensional solutions are combined to obtain the three-dimensional 19-point finite analytic scheme. Details of this process are given by Bravo et. al. (1991).

The solution of the coupled Navier-Stokes equations present an additional inconvenience; there is no clear equation for pressure. To solve this problem many methods have been developed. Notable examples are SIMPLE (Semi-Implicit Method for Pressure Linked Equations) of Patankar and Spalding (1972), SIMPLER (Patankar, 1980), SIMPLEC (Van Doormal and Raithby, 1984) and PISO (Issa et. al., 1986). Most of these methods are generally known as pressure correction methods. The program FANS-3D uses the SIMPLER and SIMPLEC methods. The discretized system of equations when used with the Finite Analytic Method can be found in the works of Bravo (1987) and Aksoy (1989).

Another major difficulty in the implementation of the numerical schemes to incompressible fluid flow problems is the choice of a proper computational grid. Clearly, it would be beneficial if one could discretize the governing equation using a grid system that places all the flow variables, scalar and vector, at the same physical location. Unfortunately, the use of such a nonstaggered grid system with a primitive variable formulation of the incompressible equations has been shown to produce nonphysical oscillations in the pressure field (Patankar, 1980). A remedy to this problem is the use of a staggered grid system; first introduced by Harlow and Welch (1965). This grid distribution was used successfully in many codes and it is still the most prevalent grid arrangement. There are many advantages of this type of staggered grid arrangement. For a typical control volume, this discretized continuity equation contains the differences of adjacent velocity components. The discretized gradient of pressure in the momentum equation also contains adjacent pressure values. This arrangement prevents the occurrence of a wavy pressure and velocity fields in the numerical solutions. In the staggered

occurrence of a wavy pressure and velocity fields in the numerical solutions. In the staggered grid arrangement, the pressure difference between two adjacent grids becomes the driving force for the velocity component at the cell face between these grid points. Besides, the mass flow rate across each cell can be calculated without any interpolation for the relevant velocity component. The staggered arrangement of the grids also eliminated the need of specifying the pressure boundary conditions on the walls.

However, the staggered grid arrangement has its disadvantages too. In this arrangement, there are two distinct cells for the application of the finite analytic method to the two-dimensional momentum equation as shown in figure 3. This implies the need to evaluate two sets of finite analytic coefficients. In the three-dimensional problems three sets of coefficients must be evaluated for the momentum equations, one for each velocity component. This increases the CPU cost and memory requirement. The case of staggered grids also increases the difficulties in programming since different geometries parameters must be used for each grid.

To solve these problems Rhie and Chou (1983) introduced a non-staggered grid arrangement. In their method all variables are evaluated at the same location, the center of each control volume. The pressure gradient in the momentum equation is still evaluated by subtracting the pressures between two non-adjacent nodes. However, to compute the continuity equation, new velocity components on the volume faces are evaluated. These velocity components are obtained by an interpolation scheme based on the momentum equation. This method apparently devised by Rhie and Chow (1983) was further study by Miller and Schmidt (1988). They called it the pressure-weighted interpolation method (PWIM). They report that the PWIM scheme predicts physically unrealistic velocities in regions of rapidly varying pressure gradients.

The first successful application of a non-staggered grid arrangement to the finite analytic method was done by Aksoy (1989). Their method, called MWIM (Momentum Weighted Interpolation Method), is similar to the original Rhie and Chow (1983) PWIM method, but with a different interpolation scheme. The advantages and disadvantages of MWIM are similar to the PWIM method. There is only one set of FA coefficients to be evaluated for three-momentum equations, reducing memory and computational time. However, using this method, unrealistic velocity components are also obtained in regions of strong pressure gradient. This problem is specially severe in coarse grid calculations.

To overcome the problem mentioned above, for the staggered and non-staggered systems, a new scheme is used in the program FANS-3D. In this new scheme, a staggered grid system is used, but only one set of coefficients is evaluated. This method uses the staggered grid arrangement, but it also uses the main concepts of the MWIM and PWIM methods. The new scheme evaluates only one set of coefficients at the center of each control volume (see figure 4). The FA coefficients at the nodes of the velocity components are obtained by linear interpolation of the coefficients obtained at the centers of the control volumes. Details of this derivation are given by Bravo (1991). This method was fully tested and it is the method used in FANS-3D.

SOME APPLICATIONS OF THE PROGRAM

Conjugate Heat Transfer in Laminar Flow Between Parallel Conducting Plates (Extended Gratz Problem)

The geometry of the problem is represented in figure 5, where fully developed flow between two infinite parallel plates is moving from left to right. The domain has a total length of $10H$ with inlet height of H . The parallel plates have a thickness equal to $0.5H$ and are kept at constant temperature T_w on their external surfaces. The flow enters at uniform temperature T_f . The conductivities of the solid and fluid are k_s and k_f . The conductivity ratio $R_k = k_s/k_f$ determines the temperature in the solid fluid interface. If the conductivity ratio R_k is very high, the temperature on this interface is very close to the external surface temperature T_w . In this case, the problem is reduced to the original Gratz problem (Bravo, 1987). In general, the distribution of temperatures is determined by the Peclet number Pe , and the conductivity ratio R_k .

The Gratz problem, when the temperature on the solid fluid interface is kept constant, was solved analytically by Prins, Mulder and Schenk (1949) and later numerically by Bravo (1987). The solution for the general conjugate heat transfer problem was later obtained by Mori, et.al.(1989). To solve the conjugate heat transfer problem, Mori et.al.(1989), represented the interfacial temperature distribution by infinite power series. They solved the governing energy equations for the solid and fluid domains using this temperature distribution as boundary condition. The Nusselt numbers, obtained by this procedure, were also presented in the form of infinite series. The major sources of error in their analysis are the truncation error during the evaluation of this infinite series and the simplification of the energy equation in the fluid flow domain. They assumed negligible diffusion in the longitudinal direction.

To compare with the results of Mori et.al. a Peclet number of 500 was selected. Four grid sizes of $10 \times 3 \times 9$, $20 \times 3 \times 20$, $40 \times 3 \times 40$ and $80 \times 3 \times 80$ along the x , y and z directions respectively were used. Insulated boundary conditions were applied on the solid inlet and outlet boundaries. The outlet boundary condition for the flow domain was considered fully developed or $dT/dx = 0.0$.

The dimensionless temperature, for this problem, is defined by

$$\theta = \frac{T - T_w}{T_f - T_w} \quad (3)$$

With this definition the dimensionless external surface temperature θ_w is equal to 0.0 and the dimensionless inlet temperature θ_f equal to 1.0. The local Nusselt number Nu_x is defined by

$$Nu_x = \frac{H q_f}{k_f(T_{fo} - T_m)} \quad (4)$$

where

q_f = heat flux at the solid fluid interface

T_{fo} = temperature at the solid-fluid interface

T_w = external constant surface temperature

T_f = uniform temperature of the incoming fluid

T_m = mixed mean temperature defined by $T_m = \frac{\int T U dA}{\int U dA}$

With $q_f = -k_f \frac{\partial T}{\partial n}$, and using the inlet height H as reference length, equation can be reduced to its dimensionless form

$$Nu_x = \frac{-1}{\theta_{fo} - \theta_{fm}} \frac{\partial \theta}{\partial n^*} \quad (5)$$

where $\theta_{fm} = \frac{\int \theta U dA}{\int U dA}$, θ_{fo} the dimensionless interface temperature and n^* the dimensionless normal.

The program FANS-3D finds the temperature θ at each nodal point. The Nusselt number is evaluated from this temperature distribution and equation (5).

Figure 6 compares the Nusselt number obtained by the program FANS-3D with the values obtained by Mori et. al. (1989) and discussed above. For the conductivity ratio $R_k = 1.0$, there is an excellent agreement between the results of Mori et. al. and the one provided by the program FANS-3D, for x values over 0.2. Below $x = 0.2$ the values given by Mori et. al. are under the ones given by the program FANS-3D. Theoretically, the Nusselt number at the entrance must go to infinity, which is correctly reproduced by the program. The lower values given by Mori et. al. are probably due to the simplified version of the energy equation used by them, which did not contain the diffusion term along the x direction. The Nusselt numbers for R_k equal to 100.0, 1000.0 and infinite are very close, indicating that for R_k over 100.0 the conjugate heat transfer problem behaves as the original Gratz problem. These values of the Nusselt number agree also quite well with the corresponding ones given by Mori et. al. (1989). They also coincide with the values computed by Bravo (1989) using the 9 point two-dimensional Finite Analytic method (these values are not shown in the figure). The program FANS-3D provides results that are reasonable and agree quite well with previous computations.

Conjugate Heat Transfer in a Compact Heat Exchanger

Another application example of FANS-3D is the compact heat exchanger shown in figure 7(a). This compact heat exchanger is similar to the class of compact heat exchangers known as offset-fin heat exchangers (Kays and London, 1984). To solve the complete compact heat exchanger is beyond the capacity of any computer, and a simplification is required. The flow between the finned plates is three-dimensional and very complex. The solution of the heat transfer problem is aggravated because of the conducting fins. The problem is a three-dimensional conjugate heat transfer problem. To obtain a solution to the problem we use the concept of fully developed flow extended to these geometries (Patankar, Liu and Sparrow, 1977). In this case, the velocity, a reduced pressure and a reduced temperature field become periodic after some entrance length. The reduced temperature is defined by

$$\theta(x, y, z) = \frac{T(x, y, z) - T_w}{T_{bx} - T_w} \quad (6)$$

where T_{bx} is the bulk temperature at any longitudinal position x . This situation is similar to the flow inside ducts of uniform cross section (Chapman, 1987). In this case, the shapes of the temperature profiles at successive streamwise locations separated by the periodic length L are assumed similar. Using this idea, it is possible to simplify the geometry of this problem and reduce the computational domain to the one represented in figures 7(b) and 7(c). An enlarged view of this small domain is shown in figure 8. The temperature in the front and back walls are

assumed constant and equal to T_w . The separation between the parallel plates is considered the reference length equal to L . The position of the fin and its dimensions are given in the same figure. The flow enters the domain in right hand side and leaves the domain in the left hand side, as shown by the arrows. The top and bottom surfaces are planes of symmetry.

Assuming constant properties, the temperature field and velocity fields are decoupled and their solution can be obtained independently. The governing equations for fluid flow are of the same type of the standard transport equation (1) and the program FANS-3D can be used without any modification.

To solve for temperature we define a new variable ϕ given by

$$\phi(x, y, z) = \frac{T(x, y, z) - T_w}{T_{bi} - T_w} \quad (7)$$

where T_{bi} is the inlet bulk temperature and T_w the external wall temperature. Using this definition and equation (6), the profiles of the dimensionless temperature at the ϕ_o at the outlet and ϕ_i at the inlet are of similar shape with the outlet condition being a constant times the inlet condition, or

$$\phi_o = \phi_i(\phi_{bo}) \quad (8)$$

where ϕ_{bo} is the bulk exit temperature. To determine the profiles of temperature at the inlet and outlet, an iterative procedure is required. We start assigning the temperature at the inlet $\phi_i(x, y) = \phi_{bi} = 1.0$, then the temperature on the outlet ϕ_o is obtained by solving the governing energy equation. Using the values of ϕ_o on the outlet the constant ϕ_{bo} is calculated by numerical integration at the outlet boundary. Finally, the temperature at the inlet ϕ_i is updated. This process is repeated until convergence is attained. All these steps are automatically performed by FANS-3D.

Figure 9 shows profiles of velocity component u along the y and z directions at the inlet and outlet planes, as obtained by the program FANS-3D. In this figure we can recognize the periodicity in the velocity distribution and the three-dimensional character of the flow. The flow not only moves up and down due to the presence of the beam, but also moves to the left and to right.

Figures 10(a) and 10(b) show the temperature profiles at three locations, at the inlet, on the beam and at the outlet. In these figures we can observe that the minimum dimensionless temperature ϕ is located on the lateral walls. The maximum temperatures are on the plane of symmetry. The temperature inside the fin, also shown in this figure, is almost constant in each cross section, but increases towards the center between the plates.

To study the effect of the fin, the pressure drop and the rate of heat transfer in this geometry can be compared with the ones for a fully developed flow between parallel plates. For example the pressure drop in the element of the heat exchanger, as computed by the program FANS-3D, is equal to -0.4565. For the equivalent situation, but without the fin the pressure drop is only -0.1200. Therefore, the increase in pressure drop due the fin is 280%.

The total increase in energy content, when the fluid moves from inlet to exit is given in Table 1. This table shows the energy increase in the element of heat exchanger for water and air and compares it with the one in a fully developed flow between parallel plates. The effect of the fin is an increase in the global rate of heat transfer of 68.7% for water and 54.2% for air. An additional computation with an infinite conductivity of the solid showed a less than 0.2% change in the rate of heat transfer. Thus, to improve the design of the compact heat exchanger, it is

possible to reduce the thickness of the fin without significantly affecting the rate of heat transfer, but reducing the pressure drop. Furthermore, the analysis in the temperature distribution shows that an staggered distribution of fins can be more effective than the regular one.

Conjugate Heat Transfer in Electronic Modules

In this section, the combined effect of convection, and radiation heat transfer is studied in an array of electronic chips displayed in figure 11. The chips are cooled by a radiatively non-participating gas flowing inside the passage. The program FANS-3D and the subroutine ANDISOR4 (Sanchez, Smith & Krajewski, 1990) were used to study the problem. The three-dimensional convection part of the problem was solved using the 19-point Finite Analytic Method and the radiation analysis using the discrete ordinates method (Sanchez, Smith & Krajewski, 1990). Results showing the effects of convection alone, combined radiation and convection, and the presence or not of a radiatively participating medium are presented.

To simplify the solution fully developed periodic flow is assumed (see discussion for the previous problem). Using this concept of periodicity along the x direction and considering some of the planes of symmetry, the problem can be reduced to the element represented in figures 11(b) and 11(c). The dimensions in centimeters of this element are given in figure 12. The fluid flow is in the positive x direction. The planes at $y=0.0$ and at $y=1.5$ are considered planes of symmetry; and the top and bottom walls are adiabatic. The inlet temperature of the fluid is considered equal to 305 K and the temperature of the blocks constant and equal to 320 K. All surfaces are assumed black. The fluid is considered radiatively non-participating transparent gas (air) with constant properties. The flow is considered laminar.

The Navier-Stokes equations are decoupled from the energy equation and the fluid field can be solved independently. The solution was obtained using the program FANS-3D with the 19-point Finite Analytic method. Once this solution was performed, the energy equation was solved using the same method explained above for the compact heat exchanger. The coupling between radiation and convection is done through energy balances on the walls. The existing radiative transfer code, ANDISORD4 (Sanchez, Smith & Krajewski, 1990), was used to solve for radiation. An S-8 implementation (80 discrete directions) of the discrete-ordinates model is applied. Although not required, the grids for the flow and radiation models are identical. When the medium is non-participating, the divergence of the radiative heat flux vector vanishes, and the transport and radiation model become explicitly decoupled. Implicitly, however, the two models are interdependent through the temperature field and the wall heat fluxes.

The results for the flow field computation are shown in figures 13 through 14. The velocity profiles for the u component are shown in figure 13 at two locations, at the inlet and the center between the blocks. From these figure, we can appreciate that the velocity between the blocks and the upper plate is similar to the velocity between two plates, or Poiseuille flow. For Reynolds number of 100 the velocities between the blocks are quite small. Figure 14 displays velocity vectors on the back plane of symmetry. In this last figure, at the center of the blocks, the flow is rotating counterclockwise. Figure 15 show velocity vectors on a plane located between the blocks in the x direction. This figure shows that some fluid is entrained from the top of the blocks and is transported to the lower sides.

Profiles of dimensionless temperature ϕ on three different planes are shown in figures 16 and 17. The temperature of the blocks is zero and the bulk inlet temperature is equal to one. Figure 16 shows these profiles when convection heat transfer alone is considered. As we expect

the profiles are perpendicular to the top and bottom insulated plates. In this figure we can also appreciate that the fluid moving between the blocks has a temperature very close to the temperature of the blocks. Specifically, the fluid near the bottom wall has almost the same temperature of the blocks. The maximum dimensionless temperature (coldest dimensional temperature) is on the top insulated wall.

Figure 17 shows dimensionless temperature profiles when convection and radiation are included. The temperature profiles are quite different from the ones discussed in figure 16. The maximum value of the temperature is in the center region between the top plate and the blocks. The top and bottom plates have an intermediate temperature between the temperature of the blocks and the temperature in the center region. The higher temperature of the top plate and the lower temperature on the bottom plate, compared with the convection case alone shown in figures 16, is because of the radiation effect. The top plate receives a net radiation coming from the blocks and the bottom plate. Because this plate is insulated, this arriving net radiant energy is dissipated by convection, which is indicated by a positive slope of the profile at this point. The ϕ profile has a maximum in the center region and decreases as we move closer the bottom plate. However, close to the bottom plate this profile increases again. This increase indicates that heat is transfer by convection from the fluid to the bottom wall. Because this wall is insulated, the same amount of energy is irradiated to the top wall.

An energy balance when convection-radiation heat transfer was considered shows an increase on the rate of heat transfer of 49.1% compared with the computation considering only convection heat transfer. In this problem then radiation is a very important and must be considered.

This final problem of electronic modules is an example of a three-dimensional conduction-convection-radiation heat transfer problem. Although conduction was not explicitly discussed, the problem was solved assuming infinite (10^{30}) conductivity in the blocks. In this situation the blocks assume a constant temperature everywhere resembling an isothermal body. The convection and convection-radiation results show the potential of the program FANS-3D in the simulation of complex three-dimensional problems that include all modes of heat transfer.

CONCLUSIONS

This study shows the solution of complex three-dimensional problems that included conduction convection and radiation modes of heat transfers with the application of the program FANS-3D. The study of these problems also shows different types of boundary conditions from the simplest boundary when the values of the variable are assigned on the boundary, to symmetric and periodic boundaries. The solutions were presented in the form of vectors and profiles given by the graphics part of the program FANS-3D. This graphic program also displays contours that include shading. The difficulties in reproducing these colors do not allow the inclusion of these pictures in this work.

Table 1 Comparison Between Energy Increase in the Element of Heat Exchanger and Energy Increase in Flow Between Parallel Plates

	Water	Air
Pr	1.76	0.704
$k(\text{W/m}^{\circ}\text{C})$	0.6775	31.27×10^{-3}
$\Delta E_{\text{with fin}}/C_1$	0.07075	0.15658
$\Delta E_{\text{parallel plates}}/C_1$	0.04194	0.10157
Inc. %	68.7	54.2

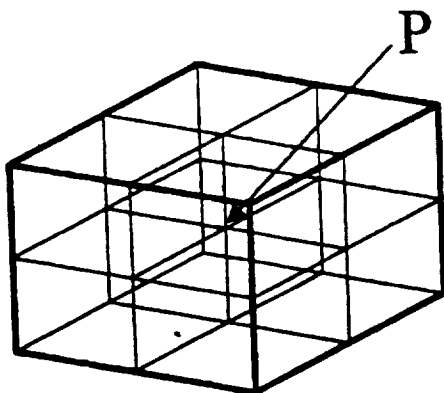


Figure 1 Finite Analytic Element

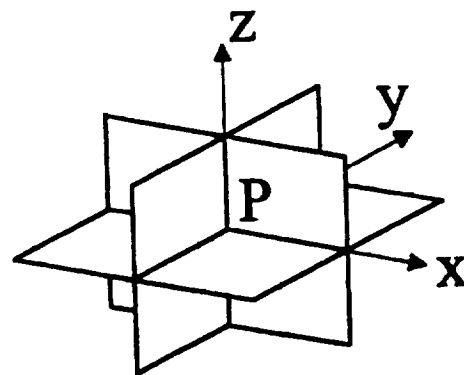


Figure 2 19-point FA Method

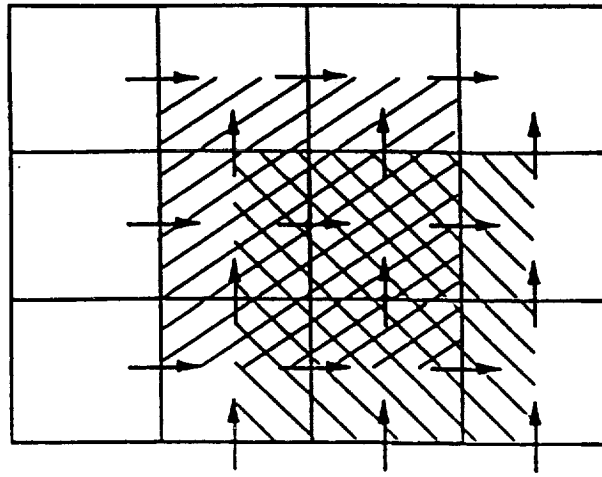


Figure 3 FA Cells in a Staggered Arrangement

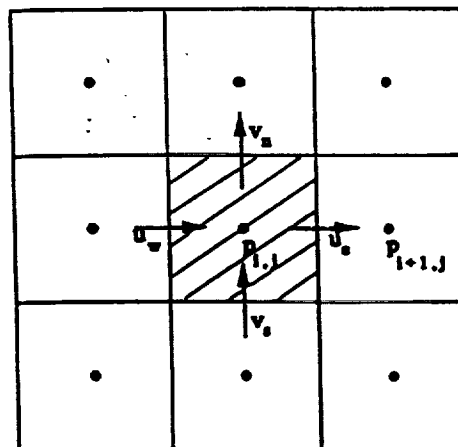


Figure 4 Evaluation of FA Coefficients
in the New Method

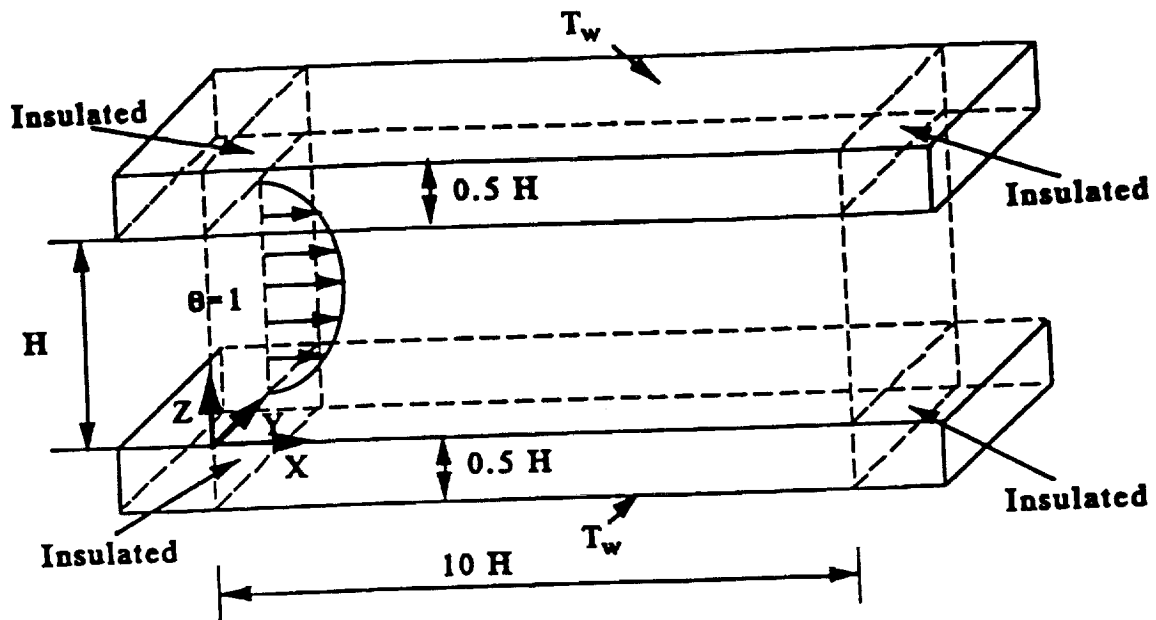


Figure 5 Domain of Solution for the Extended Gratz Problem

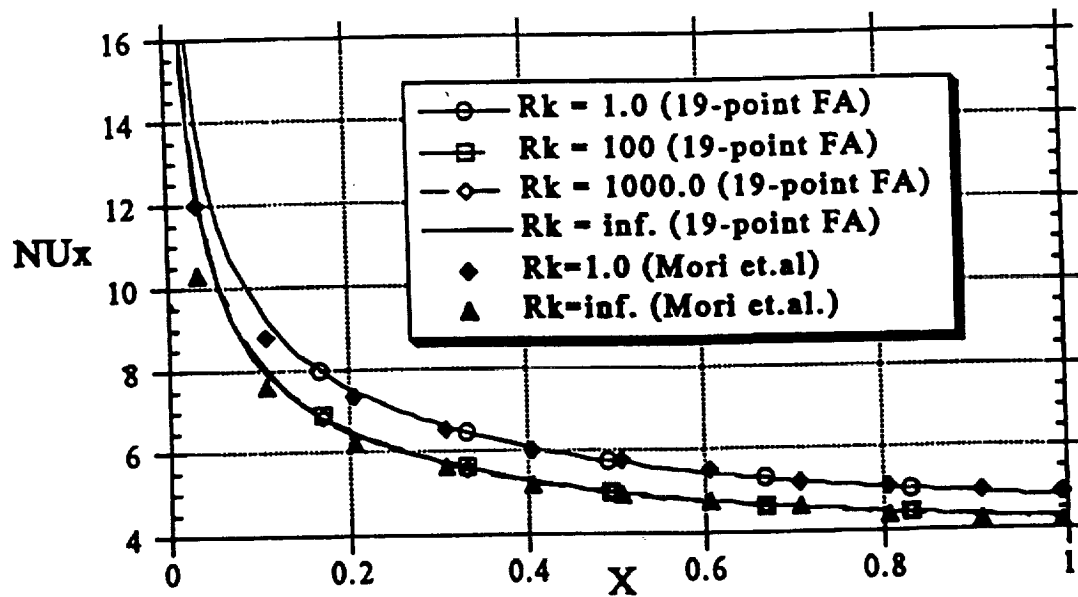
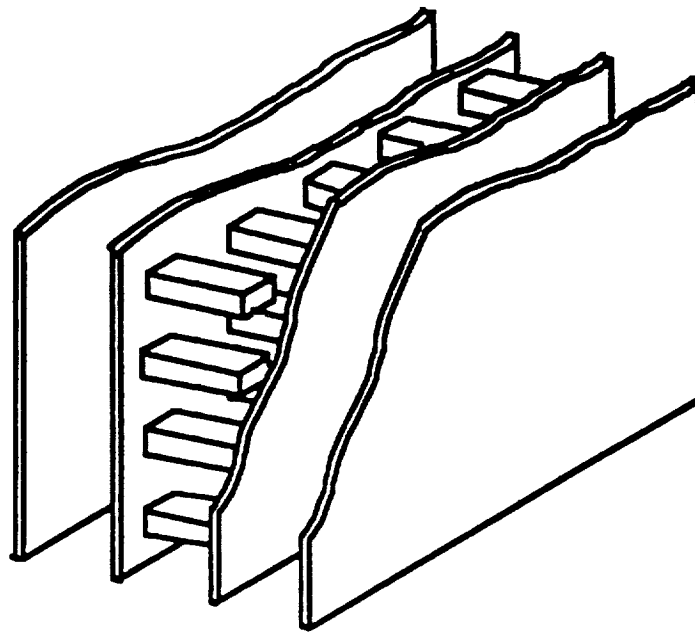
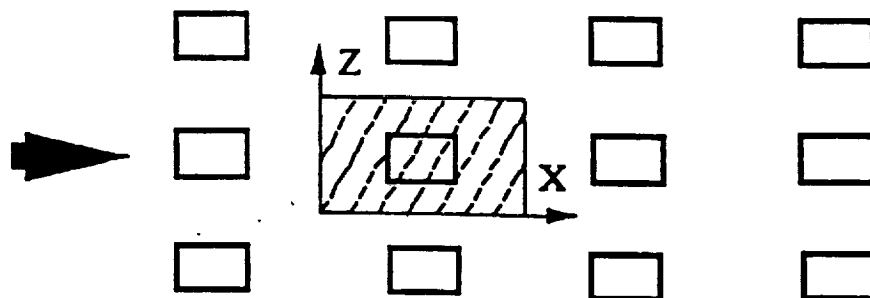


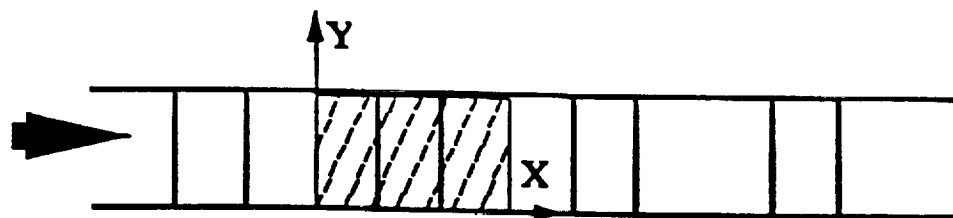
Figure 6 Local Nusselt Number at Different Locations



(a)



(b)



(c)

Figure 7 Heat Transfer Analysis in a Compact Heat Exchanger

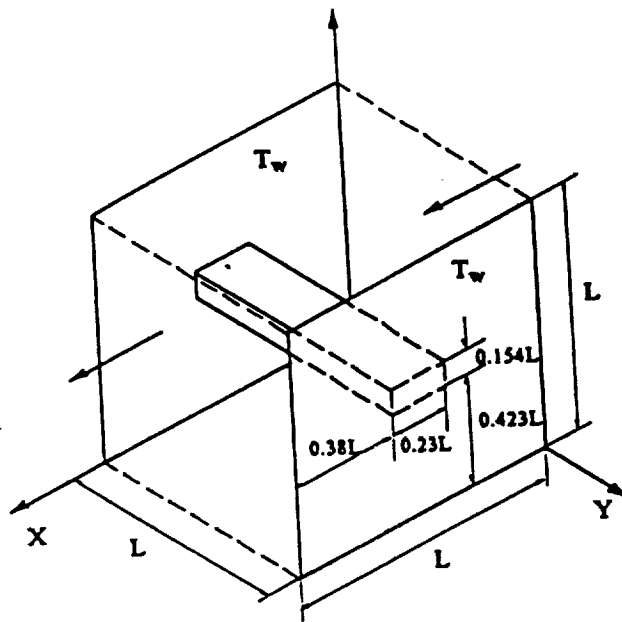


Figure 8 3-D View of the Element Shown in Figures 7 (b) and (c) of a Compact Heat Exchanger

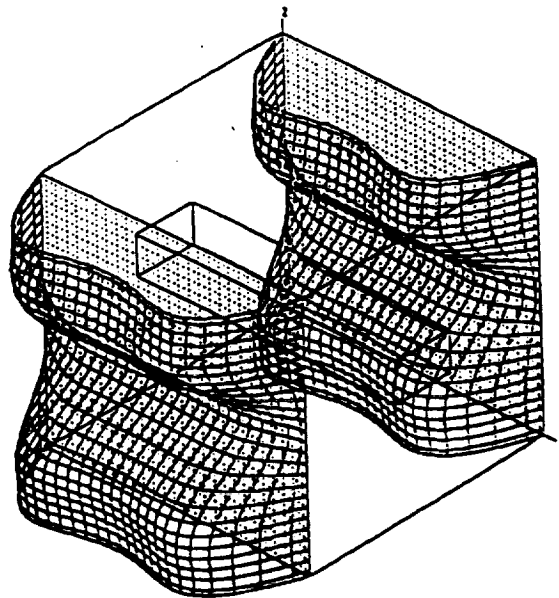
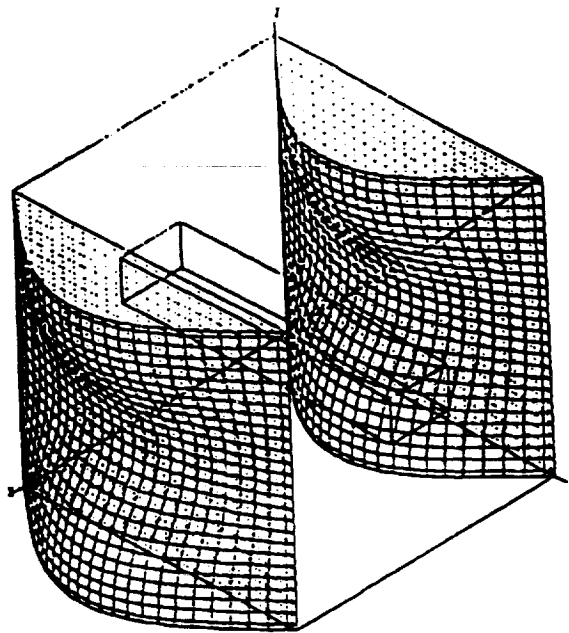
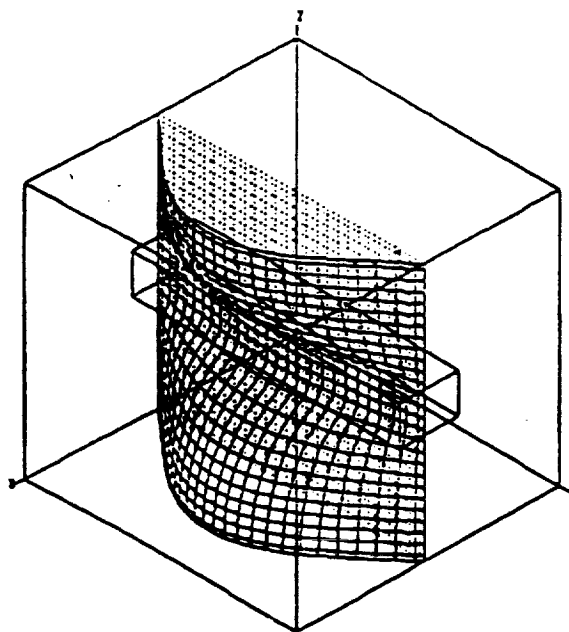


Figure 9 Profiles of U Velocity Component on the Inlet and Exit Planes. 3-D View

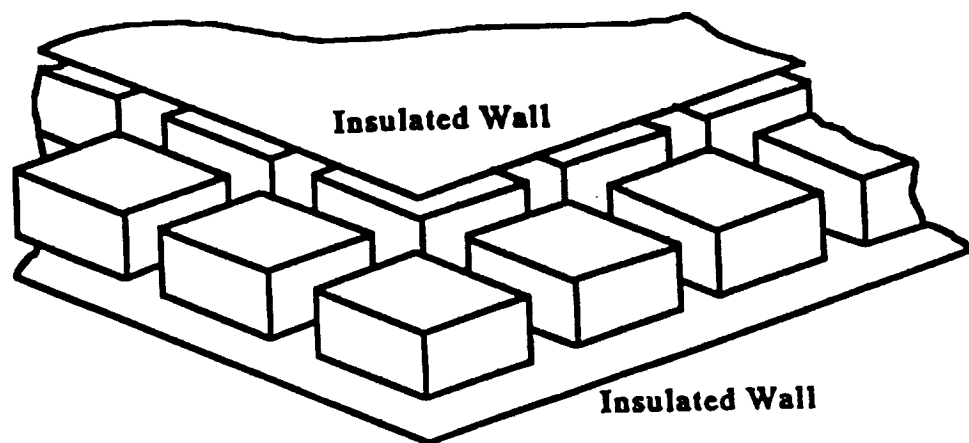


(a)

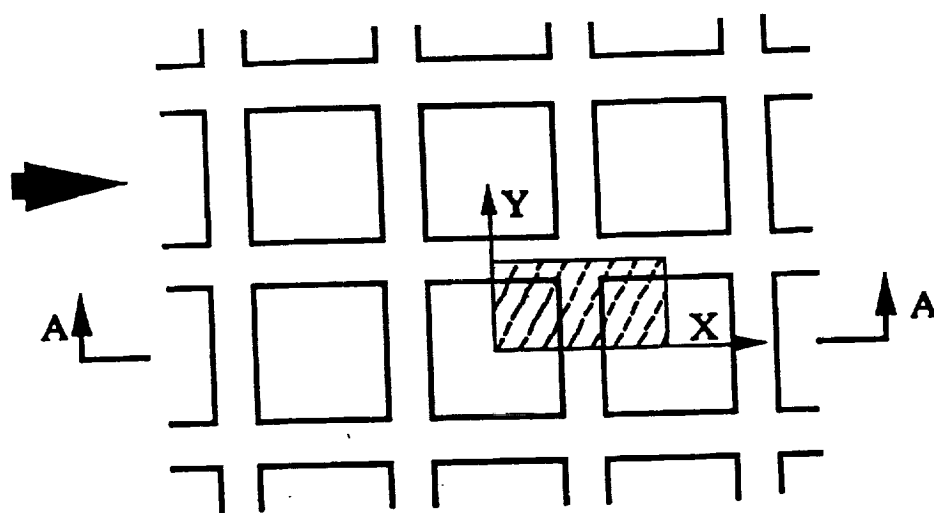


(b)

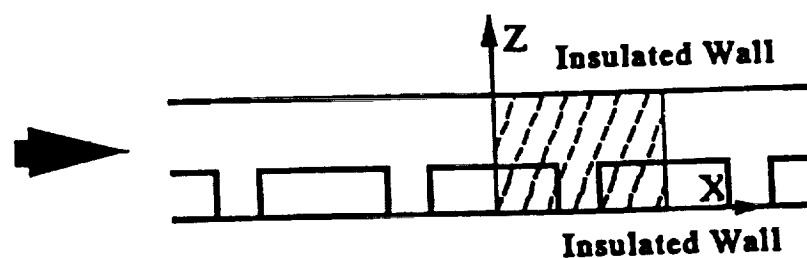
Figure 10 Temperature Profiles on Three Different Planes



(a)



(b)



(c)

Figure 11 Heat Transfer Analysis of Electronic Components

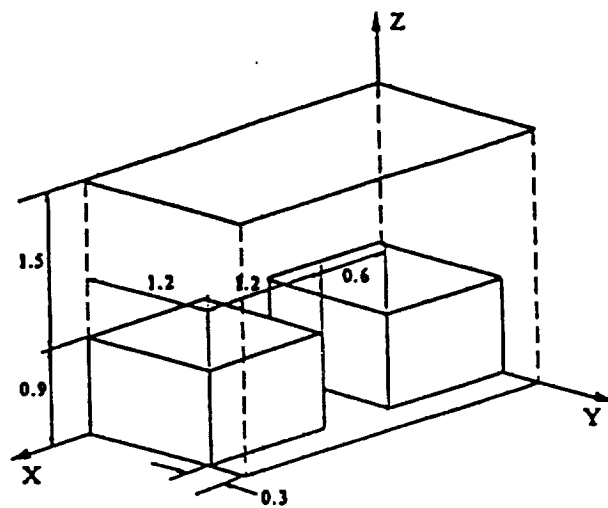


Figure 12 Domain of Solution of the Electronic Components Shown in Figure 11

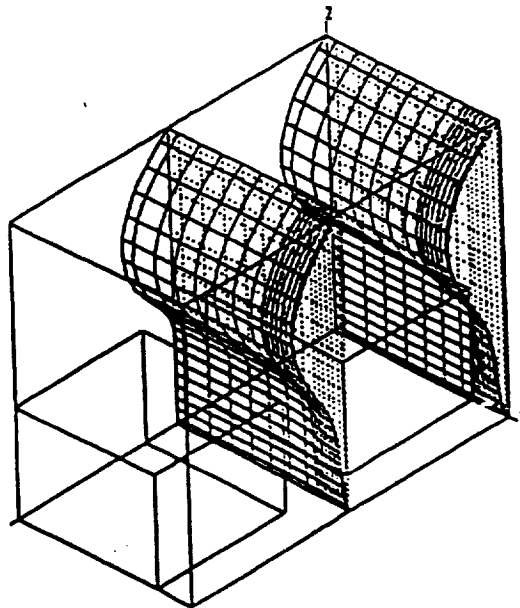


Figure 13 Profiles of Velocity Components U
3-D View

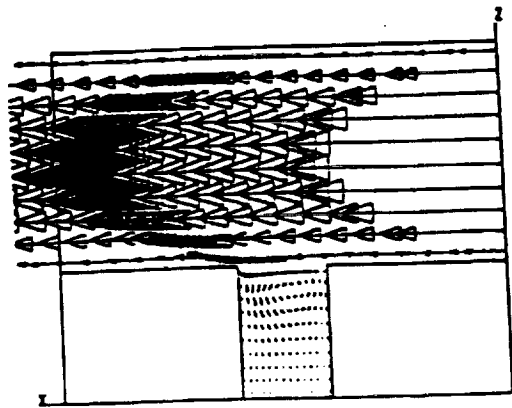


Figure 14 Velocity Vectors on Plane of Symmetry $y = 0$

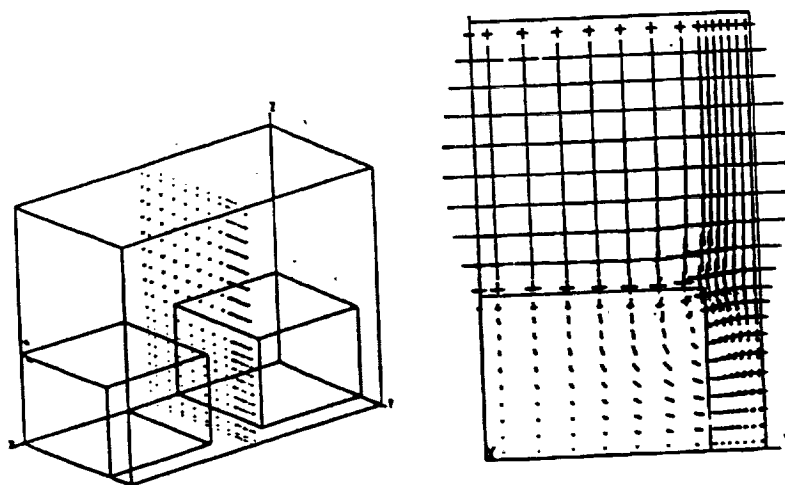
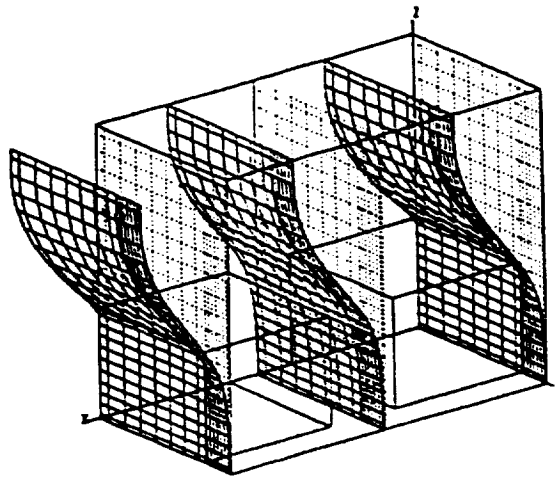
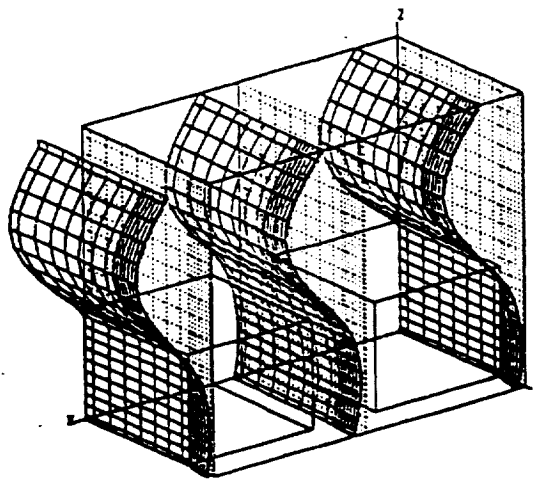


Figure 15 Velocity Vectors on Plane $x = 1.5$



**Figure 16 Dimensionless Temperature Profiles
Conduction-Convection Problem**



**Figure 17 Dimensionless Temperature Profiles
Conduction-Convection-Radiation Problem**

REFERENCES

- Aksoy H., (1989), "Finite analytic numerical solution of fluid flow and heat transfer with non-staggered grids," Master Thesis. The University of Iowa. Iowa City. Iowa.
- Bravo, R. H., (1987), "Computer-aided analysis of two-dimensional fluid flow and convective heat transfer," Master Thesis. The University of Iowa. Iowa City. Iowa.
- Bravo, R. H., (1991), "Development of the three-dimensional finite analytic method for simulation of fluid flow and conjugate heat transfer," Ph.D. Thesis. The University of Iowa. Iowa City. Iowa.
- Chapman, A. J., (1987), "Fundamentals of heat transfer," Macmillan Publishing Co. New York.
- Harlow, F. H. and Welch, J. E., (1965), "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Phys. Fluids*, Vol. 8, No. 12, pp. 2182-2189.
- Issa, R. I., Gosman, A. D. and Watkins, A. P., (1986), "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme," *J. of Computational Physics*, Vol. 62, pp. 66-82.
- Kays, W. M. and London, A. L., (1984), "Compact heat exchangers," McGraw-Hill Co. New York, Third edition.
- Miller, T. F. and Schmidt, F. M., (1988), "Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system," *Numerical Heat Transfer*, Vol. 14, pp. 213-233.
- Mori, S., Shinke, T., Sakakibara, M. and Tanimoto, A. (1979), "Steady heat transfer to laminar flow between parallel plates with conduction in wall," *Kagaku Kogaku Ronbunshu*, 1, pp. 235-240.
- Patankar, S. V. (1980), "Numerical heat transfer and fluid flow," McGraw-Hill Co. New York.
- Patankar, S. V. and Spalding, D. B. (1972), "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *Int. J. Heat Mass Transfer*, Vol. 15, pp. 1787-1806.
- Patankar, S. V., Liu, C. H. and Sparrow, E. M. (1977), "Fully developed flow and heat transfer in ducts having streamwise-periodic variations of cross-sectional area," *J. Heat Transfer*, Vol. 99, pp. 180-186.
- Prins, J. A., Mulder, J. and Shenk, J. (1949), "Heat transfer in laminar flow between parallel plates," *Applied Scientific Research*, Vol 1-2, Sec. A, pp. 431-439.
- Rhie, C. M. and Chow, W. L. (1983), "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *AIAA*, Vol. 21, pp. 1525-1532.
- Sanchez, A., Krajewski, W. and Smith, T. F., (1990), "Statistical framework for validation of satellite-based global precipitation simulation; Part I: An atmospheric radiation model - The plane layer case," Progress report prepared for Grant NA89AA-D-AC195 for National Oceanic and Atmos. Admin.
- Van Doormal, J. P. and Raithby, G. D., (1984), "Enhancements of the SIMPLE method for predicting incompressible fluid flows," *Numerical Heat Transfer*, Vol. 7, pp. 147-163.

ONE-DIMENSIONAL TRANSIENT FINITE DIFFERENCE MODEL OF AN OPERATIONAL SALINITY GRADIENT SOLAR POND

Michael C. Hicks
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Peter Golding
University of Texas at El Paso
El Paso, Texas 79968

SUMMARY

This paper describes the modeling approach used to simulate the transient behavior of a salinity gradient solar pond. A system of finite difference equations are used to generate the time dependent temperature and salinity profiles within the pond. The stability of the pond, as determined by the capacity of the resulting salinity profile to suppress thermal convection within the primary gradient region of the pond, is continually monitored and when necessary adjustments are made to the thickness of the gradient zone. Results of the model are then compared to measurements taken during two representative seasonal periods at the University of Texas at El Paso's (UTEP's) research solar pond.

INTRODUCTION

In a non-convecting salinity gradient solar pond, natural convection is artificially suppressed by establishing an internal region characterized by a strong salinity gradient. The natural buoyancy of the warmer water in the lower regions of the pond is offset by the water's higher density resulting from the higher salt concentration at the corresponding depth. As temperature of the water in the solar pond increases with depth, salt concentration also increases to the extent necessary to ensure that the density of the fluid is continuously increasing.

The stability of the solar pond is a function of the thermal and solutal diffusion rates which in turn have a strong dependence on the localized temperature of the fluid. Internal and boundary stability criteria have been developed that generally determine the necessary localized salinity gradient required to maintain a stable regime for a given temperature profile. These stability relationships have been used in conjunction with a numerical model of the transient temperature and salinity profiles to simulate the eventual erosion of the gradient zone, absent any maintenance intervention. The numerical model described in this work can be used as a predictive tool for determining the approximate shapes of the temperature and salinity profiles which can be expected over an operating period when estimated average daily ambient temperatures and insolation values are provided.

Background

Solar energy incident upon the surface of a shallow pond will be partially transmitted to the bottom of the pond where much of it is absorbed. This normally leads to natural convection as absorbed energy in the lower regions of the pond causes the fluid to thermally expand. Consequently, the thermal energy will be transported to the surface and released. In a non-convecting salinity gradient solar pond this convective process is suppressed by establishing an internal region with a salinity gradient of sufficient strength to ensure that the localized decreases in density due to thermal expansion are counter-balanced by the localized increases in density due to the higher salt concentrations. Thermal conduction will then become the dominant heat transfer release mechanism through the region of the pond where convection is suppressed. Due to the relatively low value of thermal conductivity for water (approximately $0.65 \text{ W/m-}^\circ\text{C}$) an "insulating" layer develops which allows temperatures approaching 100°C in the pond's thermal storage zone.

The typical solar pond is characterized by three regions (Figure 1); a convective surface region referred to as the Upper Convecting Zone (UCZ), a non-convective primary gradient region referred to as the Non-Convecting Zone (NCZ), and a convective thermal storage region referred to as the Lower Convecting Zone (LCZ). The thickness of each of these zones depends upon the season of the year and the manner in which the pond is being operated. For purposes of this analysis average values were derived from the research pond currently in operation at the University of Texas at El Paso (UTEP). The dimensions of this pond are 52.4 m across by 64.0 m long comprising a total surface area of $3,355 \text{ m}^2$ (about $5/6$ of an acre). The walls of the solar pond slope inward at an approximate angle of 30° from horizontal. The depth of the pond is about 3.5 meters resulting in an approximate floor area of $2,500 \text{ m}^2$.

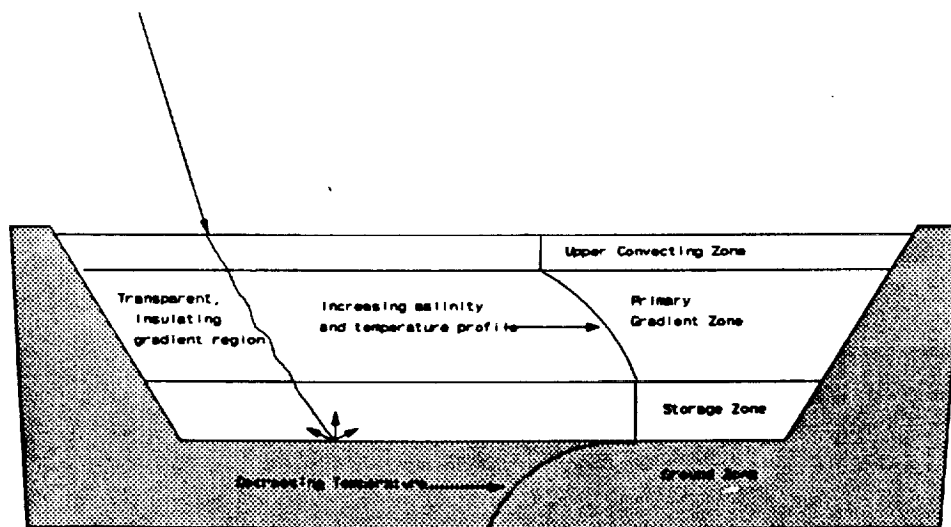


Figure 1. Schematic of a solar pond.

MODELING APPROACH

In developing a solar pond performance model a number of simplifying assumptions can be made to allow for computational ease and modeling efficiency without sacrificing the accuracy of results. This is particularly true if the modeling objective is to simulate macroscopic changes such as gradual drifts in average storage zone temperatures or changes in temperature and salinity profiles over a relatively large period of time.

Derivation of Boundary Conditions

Earlier work was performed by Hull showing that modeling for hourly variations does little to change the results obtained for the LCZ temperature over an extended period of study [1]. Since diurnal fluctuations in ambient temperature and solar insolation have a negligible impact on the solar pond's overall performance, average values of daily insolation and ambient temperature were used for the transient boundary conditions.

An interpolating function was derived for insolation on a horizontal surface from monthly averages obtained over a 21 year period for El Paso, Texas [2]. This is given as follows:

$$I_n = 6,300 + 2,300 \cdot \sin \left[\frac{2\pi(n-70)}{365.25} \right] \quad (W/m^2) \quad \text{Eq. (1)}$$

where: n = day of the year.

The portion of solar radiation incident on a horizontal surface that is reflected at the surface of the air-water interface of the pond is modeled by assuming that the incident radiation is direct beam radiation and intersects the surface at a fixed angle of incidence. These assumptions were previously reviewed by Hull in a comparison of results from a detailed computer model with an analytical model employing the assumption of direct beam radiation. The results showed that accurate estimates of the LCZ temperature can be obtained when the fixed angle of incidence (i.e., A_i) is calculated at solar noon 17 days before the autumnal equinox [1].

The following equation is used to calculate the fixed angle of incidence used in this analysis [2]:

$$\alpha_i = \cos^{-1}(\sin\beta \cdot \sin\gamma + \cos\beta \cdot \cos\gamma \cdot \cos\omega) \quad \text{Eq. (2)}$$

where: γ = latitude of El Paso, Texas (i.e., 31.5°).
 β = declination of sun on the 17th day preceding the autumnal equinox.
 ω = hour angle at solar noon (i.e., 0°).

The angle of declination (β) is the angle of the earth's axis relative to the sun-earth line and is calculated for the 17th day preceding the autumnal equinox as follows:

$$\beta = 23.44^\circ \cdot \sin \frac{(360 \cdot (\delta - 81))}{365.25} \quad \text{Eq. (3)}$$

where: δ = the day of the year for September 6th (i.e., 249)

This gives a calculated value for α_i of 25.66° which is the fixed angle of incidence used in this analysis. The associated fixed angle of refraction is calculated using Snell's Law by the following relationship:

$$\rho_F = \sin^{-1} \frac{(\sin(\alpha_i))}{n} \quad \text{Eq. (4)}$$

Using 1.333 for the index of refraction, (n), for the pond's salt water solution [3] Snell's Law yields a value of 18.99° for the fixed angle of refraction, ρ_F .

The surface reflectivity is then calculated using the following relationship derived from Fresnel's equations:

$$\rho_R = .5 \cdot \frac{\sin^2(\alpha_i - \rho_F)}{\sin^2(\alpha_i + \rho_F)} + .5 \cdot \frac{\tan^2(\alpha_i - \rho_F)}{\tan^2(\alpha_i + \rho_F)} \quad \text{Eq. (5)}$$

Applying the calculated values for the fixed angle of incidence and refraction to this equation yields an effective surface reflectivity (ρ_R) of .021.

The ambient temperature is calculated in a manner similar to that used in developing the average daily insolation. Average daily ambient temperatures are obtained from a periodic interpolating function fitted to actual historical temperatures recorded over a thirty year period by the National Oceanic and Atmospheric Administration.

$$T_a(n) = 63.55 + 19.15 \cdot \sin \frac{(2\pi(n - 105))}{365.25} \quad (^\circ\text{C}) \quad \text{Eq. (6)}$$

where: $n =$ the day of the year.

Pond Attenuation of Solar Radiation

As the radiation is transmitted through the pond it is attenuated along its path by scattering and absorption. The shorter wavelengths of the solar radiation will be transmitted through the pond with very little absorption, whereas the longer wavelengths (i.e., the infrared portion of the spectrum) will be absorbed within the first few centimeters. Wavelengths longer than the infrared wavelength (i.e., greater than 10^{-6}m) are reflected at the pond's surface.

The absorption and scattering of the solar radiation is generally represented by a transmission function from which the radiation intensity at a given depth can be determined. One such function, developed by Hull, was derived from an extensive set of experimental absorption data measured for pure water. This data was reduced into a four part summation of exponential terms.

$$\tau = \sum_{j=1}^4 (B_j) \cdot e^{-a_j x} \quad \text{Eq. (7)}$$

where: $\tau =$ the fraction of solar radiation reaching a depth (x).

This reduced function correlates well with a more extensive forty part function, derived from the same data, never deviating by more than .01 in transmittance over a 2 meter interval. The parametric values used for this transmission function are provided in Table I.

Table I Hull Four-Part Transmission Function

Exponential Extinction Coefficients for Water		
(j)	B_j	$n_j \text{ (m}^{-1}\text{)}$
1	0.237	0.032
2	0.193	0.450
3	0.167	3.000
4	0.179	35.000

When compared with measurements taken at the UTEP research solar pond on April 20, 1989 the Hull Four-Part function resulted in substantially higher transmittance values, which is to be expected since that transmission function was derived from absorption measurements in pure water. A transmission function was derived from the aforementioned measurements taken at UTEP's research pond. The following third degree polynomial provides an accurate fit to the measured data for distances ranging from 0.0 to 2.6 meters. A plot of this curve fit along with the Hull Four-Part transmission function and a third transmission function developed by Rabl-Nielson [4] have been presented in Figure 2 for comparison.

$$\tau(x) = -.066x^3 + .385x^2 - .768x + .745 \quad \text{Eq. (8)}$$

Pond Stability Criteria Considerations

The NCZ has been subdivided into two sub-regions to recognize the more stringent stability criterion that exists at the upper and lower interfaces. The internal regions of the NCZ must meet the following dynamic stability criterion [5], at a minimum, in order to ensure that thermal convection in this region is suppressed:

$$G_T < \frac{(Pr + \tau)}{(Pr + 1)} \cdot \frac{\beta}{\alpha} \cdot G_S \quad \text{Eq. (9)}$$

where:

- G_T = the temperature gradient at various locations in the NCZ ($^{\circ}\text{C/m}$),
- G_S = the salinity gradient at corresponding locations ($\%/m$),
- β = the saline expansion coefficient for NaCl (m^3/Kg),
- α = the thermal expansion coefficient ($1/^{\circ}\text{C}$),
- Pr = Prandtl number (i.e., ratio of the kinematic viscosity to the thermal diffusivity),
- τ = ratio of the saline diffusivity to the thermal diffusivity (i.e., K_S/K_T).

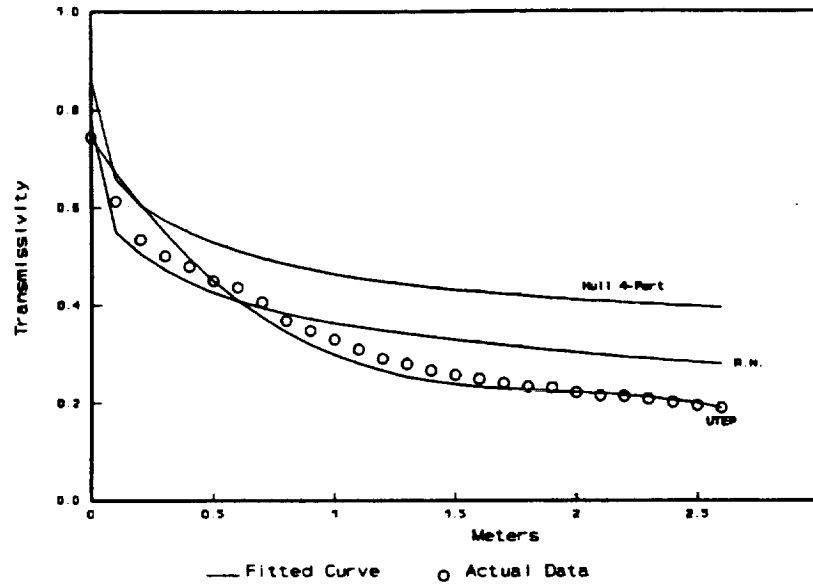


Figure 2. Comparative plot of transmission functions.

Since the values for the parameters in the above equation are dependent on both temperature and salinity it is necessary to develop a functional relationship between the salinity gradient factor (i.e., SGF, and defined as $SGF = [(Pr + \tau)/(Pr + 1)] \cdot (\beta/\alpha)$) and temperature [6]. Using information published by the Department of Interior's Office of Saline Water the following relationship was derived [7]:

$$SGF(T) = \frac{(-T^3 + 198.75^2)}{7,812.5} - (1.702)T + 48.876 \quad \text{Eq. (10)}$$

A plot of Equation 10 with the actual data for the SGF is presented in Figure 3. This relationship was then inserted into Equation 9 to obtain the necessary temperature and salinity gradient relationship that was used in establishing the internal threshold salinity gradient.

The dynamic stability criterion that is employed at both the upper and lower NCZ interfaces is based on an empirical relationship derived from Neilsen and is referred to as the Neilsen boundary condition [8]. This empirical relationship is given as follows:

$$G_s = A \cdot G_r^{0.63} \quad \text{Eq. (11)}$$

where: A is set at $28.0 \text{ (Kg/m}^4\text{)} \cdot \text{(m/K)}^{0.63}$

Development of Temperature Profile Equations

The nodal equations assume the thermal exchanges that take place between the various zones within the solar pond can be adequately represented by a transient one-dimensional model. Thermal properties of the pond are assumed to be uniform within each of the pond's zones. Using these assumptions, temperature changes, resulting from thermal exchanges within the pond, can then be represented by a

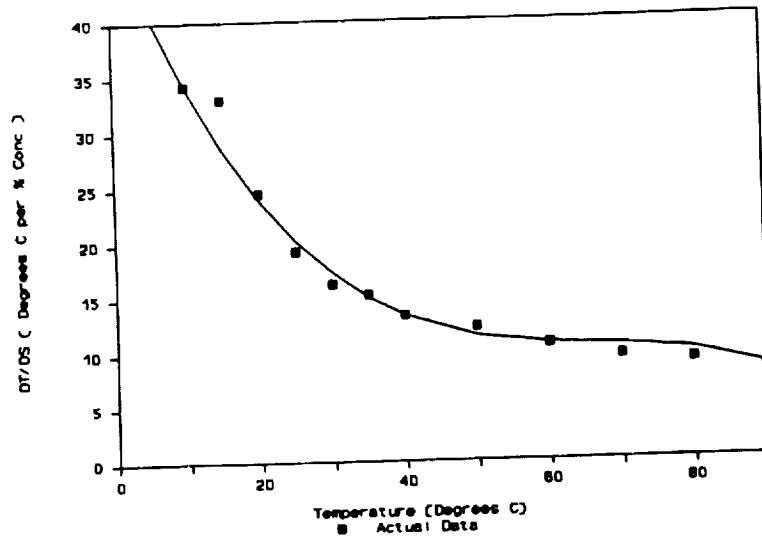


Figure 3. Comparative plot of the SGF interpolating function with actual data.

system of algebraic equations which explicitly solve for the temperature at node (i) and time (j+1) in terms of nodes (i-1, i, i+1) at time (j). These equations are derived by representing Fourier's heat conduction equation by a system of one-dimensional forward-difference equations (where x is positive in the downward direction from the surface at $x = 0.0$).

Since the UZ is essentially uniform in temperature it is represented by a single node. Its average daily temperature is set equal to the ambient air temperature (T_a) which establishes the boundary condition for the initial node. The LCZ is similarly treated by assuming no temperature gradient throughout the zone. Solar radiation reaching the depth of the NCZ-LCZ interface, and beyond, is assumed to be completely absorbed within the LCZ. A ground zone (GRZ) is modeled, in the same fashion as the NCZ, by a representative nodal network that continues to a depth where it can be reasonably assumed that a constant ambient temperature exists.

UCZ-Nodal Equation:

$$T_{UCZ} = T_a(t) \quad \text{Eq. (12)}$$

UCZ-NCZ Interface Nodal Equation:

$$T_{0,j+1} = T_{0,j}(1-2Fo-2Fo \cdot Bi) + 2Fo \cdot (T_{1,j} + Bi \cdot T_{UCZ,j}) + \frac{2Bi \cdot Fo}{h_c} \cdot (I_{1-1/2,j} - I_{1+1/2,j}) \quad \text{Eq. (13)}$$

NCZ Interior Nodal Equations:

$$T_{i,j+1} = T_{i,j} \cdot (1-2Fo) + Fo \cdot (T_{i+1,j} + T_{i-1,j}) + \frac{\delta x_n \cdot Fo}{K_n} \cdot (I_{i-1/2,j} - I_{i+1/2,j}) \quad \text{Eq. (14)}$$

NCZ-LCZ Interface Nodal Equation:

$$T_{n,j+1} = T_{n,j} \cdot (1 - 2Fo - 2Fo \cdot Bi) + 2Fo \cdot (T_{n-1,j} + Bi \cdot T_{LCZ,j}) + \frac{2Bi \cdot Fo}{h_c} \cdot (I_{n-1/2,j} - I_{n+1/2,j}) \quad \text{Eq. (15)}$$

LCZ Interior Nodal Equation:

$$T_{LCZ,j+1} = T_{LCZ,j} \cdot (1 - 2H) + H \cdot (T_{n,j} + T_{0,j}) + \frac{H}{h_c} \cdot (I_{NCZ/LCZ} + q) \quad \text{Eq. (16)}$$

$$\text{where: } H = \frac{dt \cdot h_c}{L_{LCZ} \cdot \rho \cdot C_p}$$

LCZ-GRZ Interfacial Nodal Equation:

$$T_{0,j+1} = T_{0,j} \cdot (1 - 2Fo - 2Fo \cdot Bi) + 2Fo \cdot (T_{1,j} + Bi \cdot T_{LCZ,j}) \quad \text{Eq. (17)}$$

GRZ Interior Nodal Equation:

$$T_{k,j+1} = T_{k,j} \cdot (1 - 2Fo) + Fo \cdot (T_{k+1,j} + T_{k-1,j}) \quad \text{Eq. (18)}$$

In the above equations $T_{UCZ,j}$ and $T_{LCZ,j}$ are the bulk temperatures of the UCZ and LCZ at time (j), respectively, Fo is the Fourier Modulus, and Bi is the Biot modulus. The heat transfer coefficient is set at $263 \text{ W/m}^2 \cdot ^\circ\text{C}$ and is assumed to be the same at both interfaces. This value is derived from consideration of the empirical relationship developed for two horizontal plates.

Development of Salinity Profile Equations

The salinity profile is derived from the solution of a transient one dimensional partial differential equation wherein the rate of salinity change is presented as a function of the temperature dependent solutal diffusivity (K_s) and the rate of change of salinity with respect to depth in the gradient zone. After the salinity profile is calculated for each progression in temperature profile it is checked to ensure that the stability criteria at both the upper and lower boundaries as well as within the NCZ are satisfied. If the boundary stability criterion at either the upper or lower interfaces is not met the boundary is repositioned until the stability criterion is satisfied. If the internal stability criterion is not met the run is simply terminated with the creation of a file containing the last temperature and salinity profiles. Changes to the salinity profile are based upon a numerical solution of the following differential equation describing the time dependent behavior of the salinity profile in the absence of any fresh water or brine injection into the UCZ.

$$\frac{\partial S}{\partial t} = \frac{1}{R(z)} \cdot \frac{\partial}{\partial z} \left\{ R(z) \cdot K_s \cdot \left[\frac{\partial S}{\partial z} + S_T \cdot S \left(1 - \frac{S}{\rho} \right) \cdot \frac{\partial T}{\partial z} \right] \right\} \quad \text{Eq. (19)}$$

where: $R(z) =$ the ratio of surface area at depth (z) to the surface area at the top

of the NCZ. This term only has significance in small ponds with sloping side walls.

$K_s =$ the temperature dependent solutal diffusivity of NaCl which represents the rate of molecular diffusion of NaCl in the presence of a salinity gradient.

$S_T =$ is the Soret coefficient reflecting the secondary salt transport process that occurs in the presence of a temperature gradient.

The above equation has been simplified by considering that the Soret coefficient for NaCl solutions is of the order of 1×10^{-3} 1/K to 3×10^{-3} 1/K and that for most salinity gradients it's contribution to salt transport will be relatively small. This was evidenced in analyses performed at Ohio State University, where the observed salt transport differed from the calculated salt transport, using Equation 19 without the S_T term, by less than 10% [8].

The second simplifying adjustment made to Equation 19 reflects the relative constancy of the solutal diffusivity coefficient, K_s , with variations in salinity concentrations. Generally, K_s is dependent upon both the temperature and salinity of the solution. However, for NaCl solutions the variance of K_s with salinity, over the salinity range of 0% to 30% by weight is negligible for purposes of this work. The dependence of K_s on temperature, however, results in an increase on the order of 400% over temperatures ranging from 5°C to 90°C. Therefore, a functional relationship was derived in the form of a second degree polynomial from data obtained from the Department of Interior's Office of Saline Water [7].

$$K_s(T) = (T^2 + 95.56T + 2920.9) \cdot 9.23 \times 10^{-10} \quad (m^2/hr) \quad \text{Eq. (20)}$$

Finally, the term containing information on the physical aspects of the pond walls (i.e., $R(z)$) is only important for small area ponds. Generally, the presence of sloping walls in a solar pond decreases the salinity gradient at the upper levels of the NCZ. It has been found [8] that for solar ponds with an area of 10,000 m² and a geometric factor of approximately .05 (defined as the ratio of NCZ thickness to pond length multiplied by twice the cotangent of the wall's angle with horizontal) that the maximum difference between an analysis with and without the effects of the wall slope on the salinity gradient taken into consideration is 5%. The geometric factor for UTEP's solar pond is approximately .09 (assuming a characteristic pond length of 64 m, a NCZ thickness of 1.5 m, and an angle of 300 from horizontal) which gives, from interpolation of plotted data, a maximum difference in salinity gradient of approximately 10%. The error introduced by excluding this effect is partially offset by the fact that the Soret coefficient is not included in this analysis. In light of the above considerations the diffusion rate equation can be reduced to the following;

$$\frac{\partial S}{\partial t} = K_s \cdot \frac{\partial^2 S}{\partial z^2} \quad \left(\frac{kg}{m^3 \cdot day} \right) \quad \text{Eq. (21)}$$

where: K_s is expressed in the solution as a temperature dependent parameter whose functional relationship is given by Equation 20.

This equation was numerically approximated in a manner similar to that used in developing the solution for the temperature profile. A nodal network, with internodal distances corresponding to that used in the determination of the temperature profile, was constructed with nodes placed at the upper and lower

interfaces of the GRZ. The forward difference equations for the salinity profile nodes are given as follows:

Boundary:

$$S_{n,j+1} = S_{n,j} \cdot \left[\frac{\Delta t}{\Delta X_n^2} \right] \cdot \left[\frac{\Delta X_n^2}{\Delta t} - 2K_{s_1} - K_{s_2} \right] + \left[\frac{\Delta t}{\Delta X_n^2} \right] \cdot [2(S_{n,j} \cdot K_{s_1} + (S_{n-1,j}) \cdot K_{s_2})] \quad \text{Eq. (22)}$$

Internal:

$$S_{n,j+1} = S_{n,j} \cdot \left[\frac{\Delta t}{\Delta X_n^2} \right] \cdot \left[\frac{\Delta X_n^2}{\Delta t} - K_{s_1} - K_{s_2} \right] + \frac{\Delta t}{\Delta X_n^2} \cdot [S_{n+1,j} \cdot K_{s_1} + S_{n-1,j} \cdot K_{s_2}] \quad \text{Eq. (23)}$$

UCZ Node:

$$S_{1,j+1} = S_{1,j} + (S_{2,j} - S_{1,j}) \cdot 2(K_{s_1}) \cdot \frac{\Delta t}{\Delta X_n \cdot L_{UCZ}} \quad \text{Eq. (24)}$$

where:

S	=	the salinity expressed in terms of concentration (i.e., kg/m ³)
K _s	=	the solutal diffusivity expressed in m ² /Day at the interface temperature.
K _{s1}	=	the solutal diffusivity at the upper node's temperature (m ² /Day).
K _{s2}	=	the solutal diffusivity at the lower node's temperature (m ² /Day).
L _{UCZ}	=	the upperzone thickness.

Discussion of Results

Two validation runs were performed to assess the accuracy of the computed temperature profile over a span of time. A summer month (i.e., August, 1989) and a winter month (i.e., November, 1989) were selected and actual measurements, to the extent available, were collected.

August 1989 Validation Run: Actual measurements taken at the UTEP research pond were used for the daily ambient temperature, heat extraction from the storage zone, daily solar insolation, and weekly pond profiles for temperature and salinity. The run period extended from August 2, 1989 through August 31, 1989 for a total of 30 days. A plot of the calculated temperature profile on the last day is plotted against actual temperature readings taken on that same day. As can be seen from the comparative plot, Figure 4, a close correlation of computed temperature with actual temperature exists throughout the NCZ. Only a slight variance between computed and actual storage zone temperatures exists (about .97°C or 1.3% from actual).

November 1989 Validation Run: In this run an initial temperature profile was constructed from pond measurements taken on November 7, 1989. During the run period, which spanned 22 days (from 11/7/89 through 11/28/89), the salinity gradient was not sufficiently steep at the upper and lower boundaries and an adjustment to the thickness of the primary gradient region was necessary.

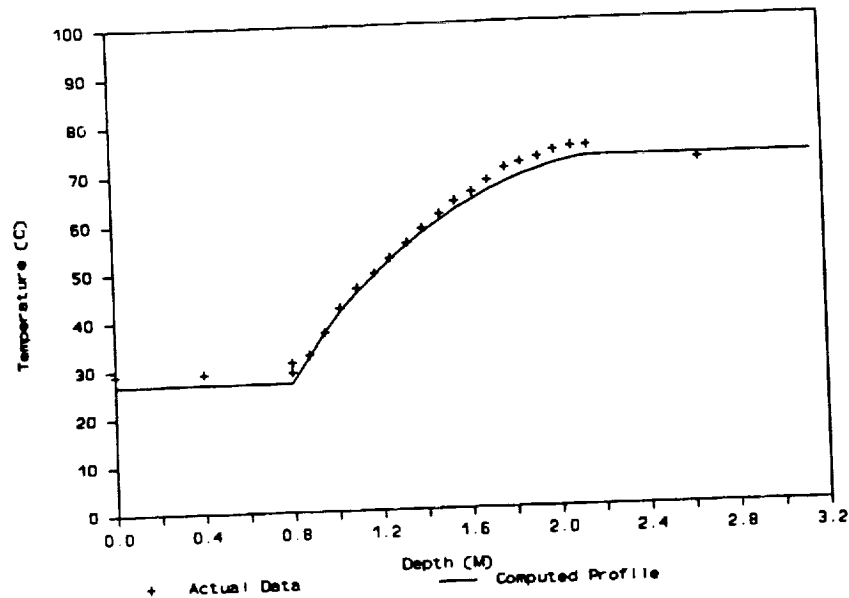


Figure 4. Comparative plot of temperature profiles for 8/31/89.

A comparative plot of the computed temperature profile on the 28th day along with actual measurements is provided in Figure 5. As with the previous August run, the results correlate well with measurements taken at the pond. The temperature variance at the mid point in the storage zone is 5.0°C (i.e., approximately 10.0% of actual). This higher variance is primarily due to the fact that the LCZ exhibited a pronounced temperature stratification (which was not modeled) during this period of time. The resulting salinity profile, derived from this temperature profile, is presented in Figure 6. As previously mentioned, a boundary adjustment occurred at both the upper and lower NCZ interfaces in order to satisfy the boundary stability criterion.

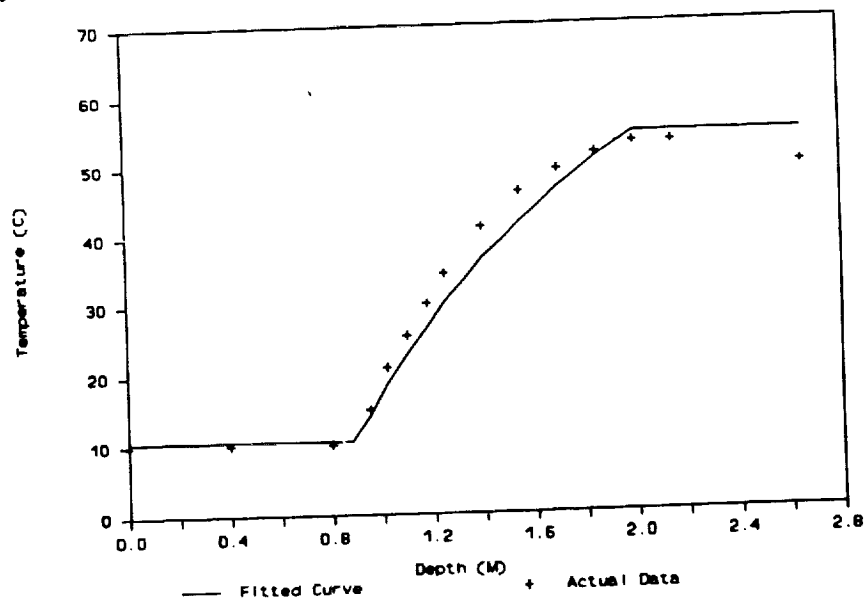


Figure 5. Comparative plot of temperature profiles for 11/28/89.

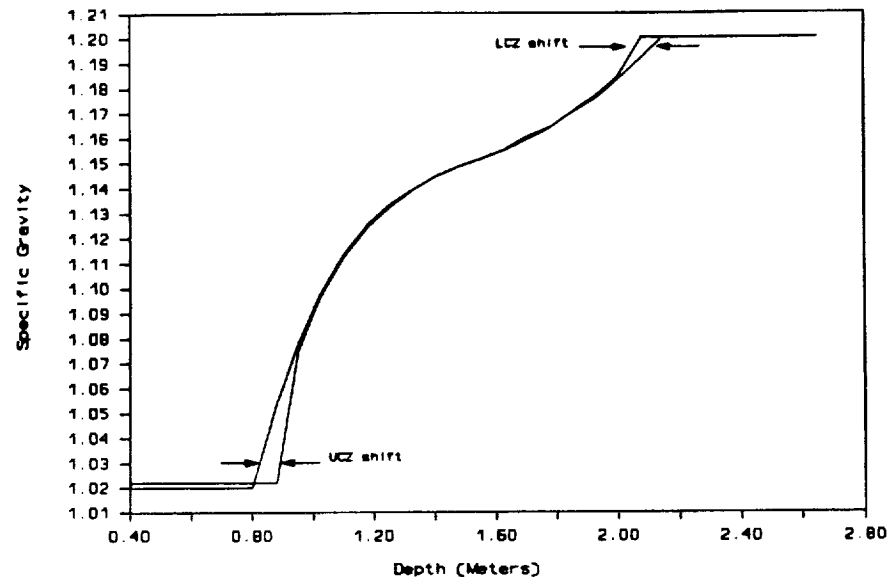


Figure 6. Transition in salinity profile resulting from shifts in NCZ boundaries.

CONCLUSIONS

The double diffusive transfer processes typical of a salinity gradient solar pond was represented by a system of coupled finite difference equations to simulate the interdependency of the ponds temperature and salinity gradients. A number of simplifying assumptions can be readily made to allow for an easily generated solar pond numerical model that will provide a reasonable first order prediction of changes in temperature and salinity profiles, bulk temperature changes in the ponds storage zone, and changes in the thickness of the gradient zone. This modeling approach may be used for predicting attainable storage temperatures, gradient zone maintenance schedules, and salinity concentration requirements. It is applicable to a variety of conditions for purposes of determining the economic viability of a new solar pond and could be employed in the operational planning of currently operating solar ponds. The results also provide valuable insight into the salinity profiles which need to be installed in order to maintain a stable thermal stratification in an established solar pond (for a given storage zone temperature associated with a given application). Further, the time variance calculation of the profiles provides useful predictions of the salinity gradient modifications which need to be accomplished during the time evolution of the operating pond in order to sustain operation.

REFERENCES

1. Hull, J. R. : Computer Simulation of Solar Pond Thermal Behavior. Solar Energy, vol. 25, 1980, pp. 33-40.
2. Richards, L. M. and Bush, G. E. : Solar Geometry and Time. Solar Energy Technology

Handbook, 1980, pp. 39-65.

3. Weinberger, H. : The Physics of the Solar Pond. Solar Energy, vol. 8, 1963, pp. 45-56.
4. Knudsen, J. D. : Numerical Simulation of Large Scale Solar Ponds. Master Thesis for University of Texas at El Paso, 1987.
5. Elwell, D. L., Short, T. H., and Badger, P. C. : Stability Criteria for Solar (Thermal-Saline) Ponds. Journal Series No. 68-77, Ohio Agricultural Research and Development Center.
6. Hicks, M. C. : Computer Performance Model of a Solar Pond-Coupled Desalination System. Master Thesis for University of Texas at El Paso, 1990.
7. Saline Water Conversion Engineering Data Book. Office of Saline Water, 2nd Ed., United States Department of the Interior, 1971.
8. Hull, J. R., Nielsen, C. E., and Golding, P. : Salinity-Gradient Solar Ponds. CRC Press.
9. Akbarzadeh, A. : Effect of Sloping Walls on Salt Concentration Profile in a Solar Pond. Solar Energy, vol. 33, 1984, pp. 137-141.

AEROSPACE APPLICATIONS OF SINDA/FLUINT AT THE JOHNSON SPACE CENTER

Michael K. Ewert
NASA Johnson Space Center
Houston, Texas

Phillip E. Bellmore
McDonnell Douglas Space Systems Co.
Houston, Texas

Kambiz K. Andish, John R. Keller
Lockheed Engineering & Sciences Co.
Houston, Texas

ABSTRACT

SINDA/FLUINT has been found to be a versatile code for modeling aerospace systems involving single or two-phase fluid flow and all modes of heat transfer. The code has been used successfully at the Johnson Space Center (JSC) for modeling various thermal and fluid systems. Features of the code which have been utilized include transient simulation, boiling and condensation, two-phase flow pressure drop, slip flow, multiple submodels and depressurization. SINDA/FLUINT has been used at JSC to support the Space Shuttle, Space Station Freedom and advanced programs. Several of these applications of SINDA/FLUINT at JSC are described in this paper.

SINDA/FLUINT is being used extensively to model the single phase water loops and the two-phase ammonia loops of the Space Station Freedom active thermal control system (ATCS). These models range from large integrated system models with multiple submodels to very detailed subsystem models. An integrated Space Station ATCS model has been created with ten submodels representing five water loops, three ammonia loops, a Freon loop and a thermal submodel representing the air loop. The model, which has approximately 800 FLUINT lumps and 300 thermal nodes, is used to determine the interaction between the multiple fluid loops which comprise the Space Station ATCS.

JSC has also developed several detailed models of the flow-through radiator subsystem of the Space Station ATCS. One model, which has approximately 70 FLUINT lumps and 340 thermal nodes, provides a representation of the ATCS low temperature radiator array with two fluid loops connected only by conduction through the radiator face sheet. The detailed models are used to determine parameters such as radiator fluid return temperature, fin efficiency, flow distribution and total heat rejection for the baseline design as well as proposed alternate designs.

SINDA/FLUINT has also been used at JSC as a design tool for several systems using pressurized gasses. One model examined the pressurization and depressurization of the Space Station airlock under a variety of operating conditions including convection with the side walls and internal cooling. Another model predicted the performance of a new generation of manned maneuvering units. This model included high pressure gas depressurization, internal heat transfer and supersonic thruster equations. The results of both models were used to size components, such as the heaters and gas bottles and also to point to areas where hardware testing was needed.

INTRODUCTION

Use of the Systems Integrated Numerical Differencing Analyzer (SINDA) analysis tool has expanded steadily over the years since its origin in the 1960's. The Fluid Integrator (FLUINT) code added significant fluid system analysis capabilities under a NASA Johnson Space Center (JSC) contract in 1985 (ref. 1). Since then, analysis applications for the code have increased in scope as well as in number. At the Johnson Space Center, SINDA/FLUINT has been used to solve both steady state and transient thermal/ hydraulic problems involving single and two-phase fluid flow with heat transfer. Conduction, convection and radiation have been modeled in simple and complex systems alike. Models have ranged from a few nodes to a few thousand nodes. SINDA/FLUINT has been used at JSC to support the Space Shuttle, Space Station Freedom (SSF) and other advanced technology programs. The code has been used in the conceptual design, detailed design, test and performance verification phases of these programs.

The Space Station Thermal Control System (TCS) consists of both passive thermal control features and active fluid loops to transport heat. The power system uses a single-phase ammonia cooling system and the habitable modules use single-phase water loops to gather heat from racks filled with equipment. The internal thermal control system (ITCS) water loops transfer their heat to one of three external thermal control system (ETCS) two-phase ammonia loops which dump the heat to space. Figure 1 shows a schematic representation of the Permanently Manned Capability (PMC) Space Station Freedom and indicates the location of some of the TCS features. Thousands of SINDA/FLUINT CPU hours have been logged analyzing the various Space Station Freedom Thermal Control Systems during the design phase, and this use will continue into the operational phase since SINDA/FLUINT is the official code used for Space Station Integrated Thermal Analysis. SINDA/FLUINT submodels developed by various system designers are integrated and run at JSC in order to analyze integrated system performance under various nominal and off-nominal conditions.

Other more detailed SINDA/FLUINT models have been built of particular components of the SSF Active Thermal Control System (ATCS), particularly the ETCS radiators. These radiators have undergone several redesigns in order to improve performance and reduce weight, cost and assembly time. SINDA/FLUINT has been used in these design trade studies in order to evaluate potential alternate designs. One such study, which will be described here, looked at ways to prevent the radiators from freezing when subjected to an environment temperature substantially below the freezing point of the ammonia working fluid.

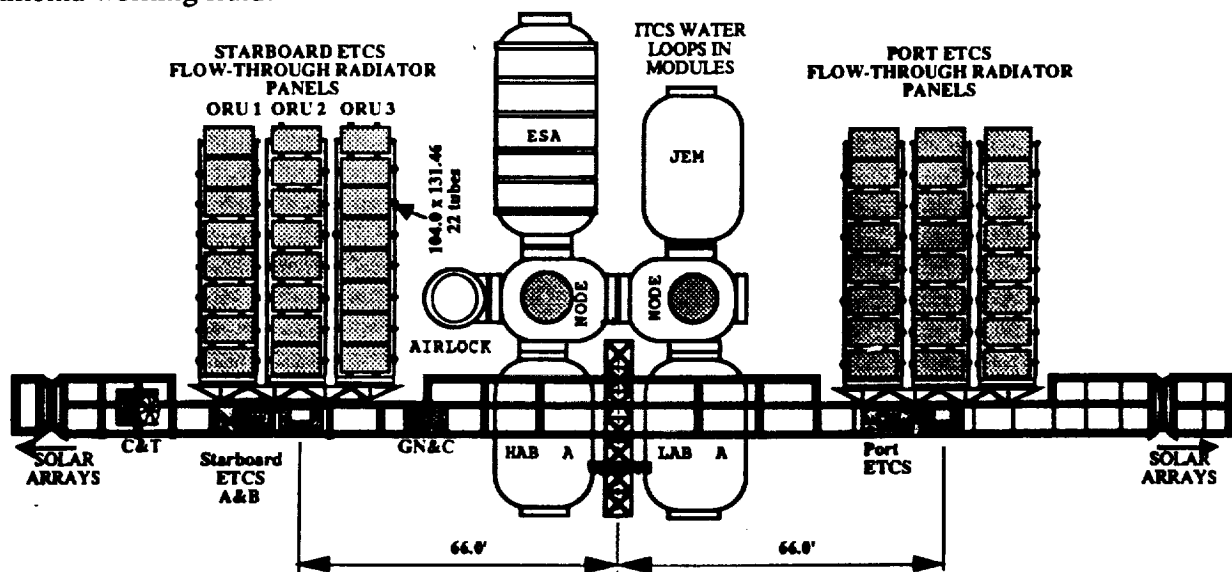


Figure 1: PMC Configuration of Space Station Freedom

Ideal gas systems are another area in which SINDA/FLUINT has been used successfully. Using the "volume flowrate set" (VFRSET) feature in SINDA/FLUINT to model a vacuum pump, the Space Station airlock has been modeled during depressurization and hyperbaric operations. Other features of the code have been used to model the storage tank and thrusters of a new manned maneuvering unit.

SPACE STATION FREEDOM INTEGRATED THERMAL ANALYSIS

SINDA/FLUINT has been utilized in support of Space Station Freedom Integrated Active Thermal Analysis. This task is responsible for integrating SINDA/FLUINT math models of all space station active thermal control systems and analyzing the resultant math model under a variety of conditions. Analyzed conditions include nominal and off-nominal operations.

Analysis of nominal operations generally consists of utilizing a time variant applied heat load and environmental conditions and determining if the system response is acceptable. Off-nominal analysis would consist of such conditions as pump failures, control system failures, imbalances between applied and rejectable heat loads, and system startups/shutdowns. The SINDA/FLUINT models have been used to analyze all of the above transients. The SINDA/FLUINT models will be briefly described and examples of problems solved will be given.

SINDA/FLUINT Model Descriptions

The SINDA/FLUINT models utilized consist of two basic types - the models of the internal habitable modules (internal thermal control system - ITCS) and the models of the external two-phase ammonia heat transport system (external thermal control system - ETCS). The models of the internal systems consist of the United States resource nodes, the United States habitation and laboratory modules, the Japanese experiment module, and the European Space Agency Columbus module. For all modules, both a detailed and a simplified version of the models have been developed.

The modeling philosophy utilized for the detailed ITCSs is to provide sufficient resolution so that the exit temperature of each experiment rack or equipment rack can be determined and the majority of conceivable configurations and transients can be accurately modeled. The transient response time is also considered in order to accurately predict the energy transport to the ETCS. The models do not consider the temperatures of the individual coldplates within a rack, of which there can be from 4 to 8 in each rack. The nodalization for detailed models typically consists of 150 to 200 FLUINT nodes. These models are used to evaluate system cooling, controls stability, pump sizing, and off-nominal performance.

For the simplified ITCS models, the response desired is the overall module response to total module heat load changes. The simplified models are typically used to provide more realistic boundary conditions for the ETCS. Therefore, the nodalization can typically consist of fewer than 20 FLUINT nodes.

United States Habitation and Laboratory System and Model Description

The U.S. habitation (U.S. Hab) module is the primary living quarters for the Space Station occupants, while the laboratory (U.S. Lab) module is the primary U.S. location for research. Both modules utilize an ITCS architecture that nominally acts as a two-loop system but is capable of being reconfigured as a one-loop system (Figure 2). When in two-loop mode, the section connected to the low temperature interface heat exchanger (IHx) is termed the LT branch and the section connect to the moderate temperature IHx is termed the MT branch. The two-loop to one-loop architecture is prevalent onboard the Space Station, being utilized by all U.S. modules and currently being considered by the International Partners.

The U.S. Hab/Lab system uses a System Flow Control Assembly to maintain a constant pressure drop between the rack supply and the rack return lines. The maintenance of the constant "delta-p" helps ensure adequate flow to each rack. The flow to a particular rack is controlled by the Rack Flow Control Assemblies which monitor either temperature or flowrate, depending on the particular type of control desired. Variation of a rack's flowrate is necessary to conserve power since not all racks will require cooling at all times. The U.S. Hab/Lab system also includes a regenerative heat exchanger that is used to ensure that the water entering the MT branch is above 60°F (15.5°C). The 60°F setpoint was chosen since the Space Station maximum allowable dewpoint is 60°F, thus precluding condensation on the MT branch of plumbing and saving the expense of insulating the entire system. Each Interface Heat Exchanger (IHX) has a bypass line that is used to maintain a desired overall IHX outlet temperature.

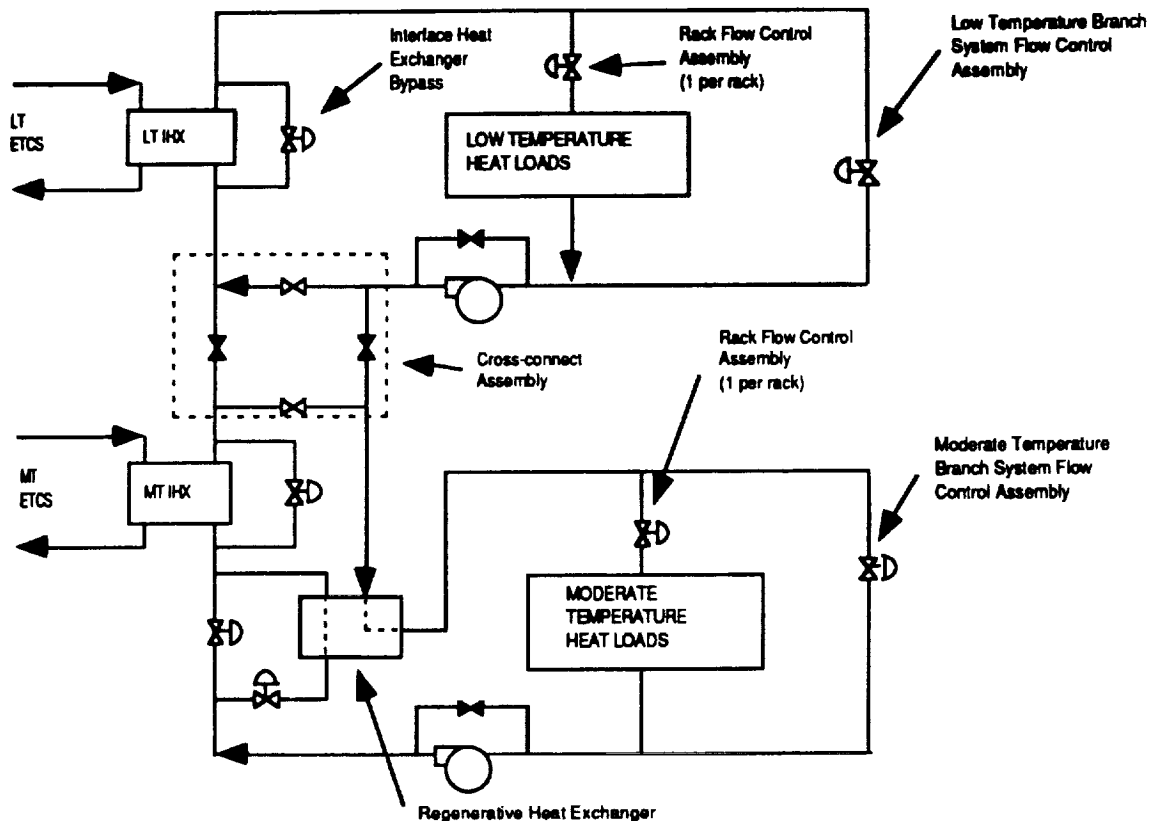


Figure 2. United States Habitation/Laboratory Functional Schematic

The FLUINT models of the habitation and laboratory modules are similar, the primary difference being the number of payload and system racks. The level of detail is such that all racks are modeled as well as important control systems. Figure 3 provides a schematic showing the number and location of the FLUINT components for the U.S. Laboratory module.

For the U.S. Lab model, most fluid line lengths and sizes have been defined and have been incorporated. The locations of elbows and some connections have not yet been defined. The control valve characteristics, such as k-loss and flow areas, have been defined for the Rack/System Flow Control valves. The characteristics for the remainder of the valves are not yet defined. The Rack/System Flow Control valve is a sculpted ball valve with a variable k-loss. This action is modeled within FLUINT by altering both the valve angle and the k-loss value. The thermal mass of the equipment in each rack is accounted for by including a SINDA thermal node equivalent to 30 lb_m (13.62 kg) of stainless steel, though the actual mass of equipment in each rack is currently unknown.

The fluid pump head curve has been defined and is incorporated into the FLUINT model. The FLUINT pump model is strictly valid only in the first quadrant of operation, i.e. windmilling, reverse flow, etc., can not be predicted. The model can predict changes in pump operation due to changes in speed.

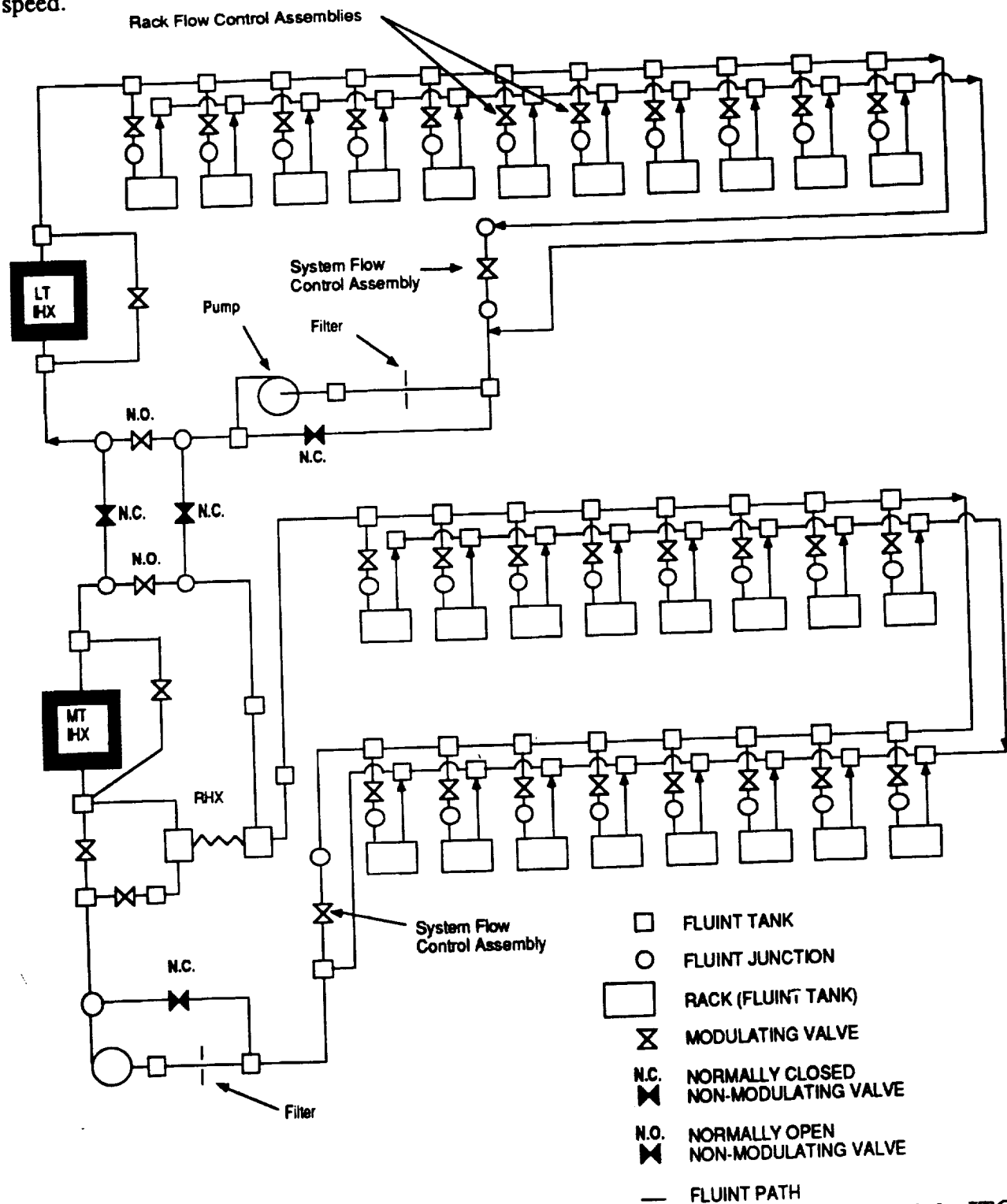


Figure 3. FLUINT Nodalization of the United States Laboratory Module ITCS

The model of the U.S. Hab maintains the same level of detail, though the number of FLUINT components is reduced since the U.S. Hab has fewer racks.

Resource Node System and Model Description

The resource node contains the primary command and control avionics for the Space Station. The resource node ITCS cools the resource node 2, resource node 1, airlock and Pressurized Logistics Module (PLM). The PLM can be located at either resource node 1 or resource node 2. Since the airlock, resource node 1 and PLM are attached at different stages of the Space Station construction, and possibly at different locations for the PLM, several distinct configurations of the resource node ITCS exist. This variability forces special modeling considerations for the resource node and will be discussed below. The architecture of the resource node is the most developed at this time since the resource node will be the first habitable module launched. Therefore, the FLUINT model of the resource node is the most complex and also the most useful for predicting ITCS performance.

The ITCS for the resource node is a simplified variant of the ITCS for the U.S. laboratory module discussed above. Unlike the U.S. Hab/Lab systems, the equipment in the resource node is normally always on. Therefore, the need to provide a variable flowrate is not present. The ITCS developers have chosen to use an orifice to control the flow to each rack with the coolant pumps set to a constant speed. There are some active components associated with the resource node, the primary ones being rack flow control valves associated with the PLM. These valves will modulate to maintain a constant exit temperature for each PLM rack. Also associated with the PLM is a modulating delta-pressure valve which acts to maintain a constant pressure for the inlet header to the PLM racks.

The detailed FLUINT model of the resource node uses the same component descriptions and modeling as the U.S. Hab/Lab due to common hardware. Such items as IHX bypass valves, RHX modeling, and pump modeling are identical. The modeling of the rack is also the same. Modeling specific to the resource node involves the level of detail of the plumbing and the variability in the configuration.

The geometry of the resource node is established sufficiently such that three-dimensional isometric drawings, with dimensions, have been generated. These drawings allow for a more accurate definition of ITCS line length. Also, the pressure drop effect of elbows and bends can be estimated using the effective length approach. Though the FLUINT model does not yet consider the effect of fluid connectors, such as tees and quick disconnects, the actual connectivity of the ITCS can now be included into the model, with the pressure drop data included when available.

The variability of the resource node configuration results in four unique configurations, not counting the position of the PLM. The multiple configurations of the ITCS are accounted for by utilizing the capability of FLUINT to include sections of input from predefined external files. Each module is contained in a separate input file and included as necessary to create the ITCS connectivity of the particular configuration. The use of external files allows for easy maintenance of the various portions of the resource node ITCS model. Should a particular module change, only the model of that module need be altered to make the change effective for all configurations. The alternative would be to make changes to as many as eight separate models, which could lead to errors in implementation.

International Partners System and Model Description

The International Partners, the European Space Agency (ESA) and the National Space Development Agency of Japan (NASDA), are currently in the process of re-designing their respective ITCSs. The baseline design for both is a one-loop system with temperature controlling bypasses. The re-design efforts are due to problems in meeting Space Station thermal load management goals. The final configuration will not be known for several months, however some of the basic components will probably still exist.

The ESA ITCS is similar in concept to the resource node ITCS. The coolant flow to each primary payload rack is controlled via an orifice. In parallel with the racks is an avionics heat exchanger. The

outlet temperature of the avionics heat exchanger is controlled to a setpoint by a modulating valve. The operation of the valve will cause pressure variations at the inlet of the heat exchanger and also the payload racks. The coolant pump's speed is controlled to maintain a constant pressure at the inlet of the racks.

The NASDA ITCS contains several dedicated cooling loops which interface with the main cooling loop via heat exchangers. Of particular interest is a Freon cooling loop which services an external platform. The ability to model multiple fluids, while not unique to FLUINT, enhances the range of problems that FLUINT can be applied to. The NASDA ITCS has a combination of orificed and actively controlled payload rack locations. Pump speed is controlled based both on pressure and temperature inputs.

Models of several proposed International Partners ITCSs have been created and utilized to perform analysis. The analysis has concentrated on verifying the general acceptability of proposed design changes and will be discussed in more detail below. SINDA/FLUINT has shown to be a very versatile and well-behaved analysis tool during the course of these numerous modifications.

External Thermal Control System and Model Description

The External Thermal Control System (ETCS) is the primary method for transporting the waste heat generated within the habitable modules for ultimate rejection to space. The ETCS is a two-phase ammonia system with several unique components such as the Rotary Fluid Management Device (RFMD), Back Pressure Regulating Valve (BPRV), and cavitating venturies. Additional components include two-phase ammonia to single-phase water heat exchangers, condensing radiators, a bellows accumulator, two-phase coldplates, and associated plumbing. The ETCS has three separate cooling loops, two Low Temperature (LT) loops operating from 33 to 39°F (1 to 4°C) and one Moderate Temperature (MT) loop operating from 55 to 62°F (13 to 17°C). Figure 4 provides an ETCS functional schematic for the Man Tended Configuration of SSF.

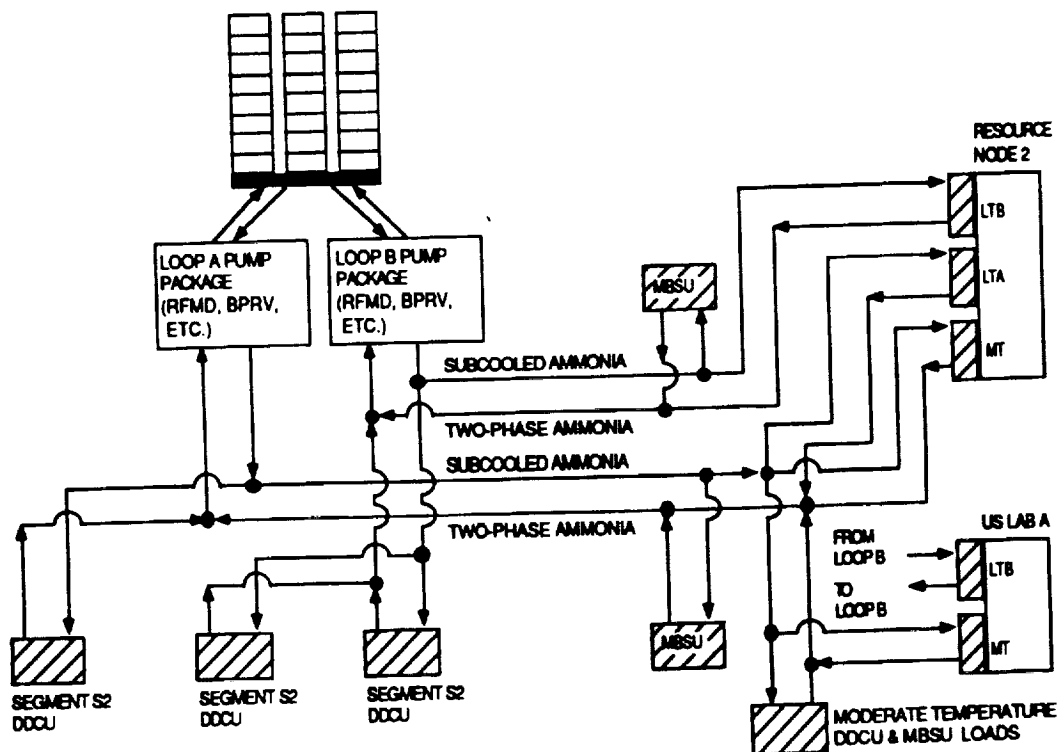


Figure 4. ETCS Functional Schematic for the Man Tended Configuration

The system operation consists of the RFMD supplying slightly subcooled ammonia liquid to a distributed system of heat exchangers. The flow to each heat exchanger, and therefore the ultimate heat acquisition capability of each heat exchanger, is controlled by a cavitating venturi placed at the inlet of each heat exchanger. The downstream pressure is controlled so that the ammonia will be very close to the saturation point after exiting the cavitating venturi. With the addition of heat from the internal modules, the ammonia begins to boil, enhancing the heat transfer process. The heat transferred to the ammonia is controlled so that the maximum ammonia vapor quality is no greater than 90%. The resulting two-phase mixture returns to the RFMD.

The RFMD consists of a stationary outer housing with a rotating inner housing that acts to separate the two-phase mixture into vapor and liquid components (Figure 5). The ammonia vapor is plumbed to the condensing radiators where the acquired heat is rejected to space. The condensate then returns to the RFMD where its temperature is increased and then pumped back out to the module's heat exchangers. The pumping action of the RFMD is via immersed pitot probes which convert the rotational energy into a static pressure rise, thus providing the necessary pumping power. The RFMD maintains an internal liquid level by means of a level pitot which acts to pump excess fluid to the bellows accumulator.

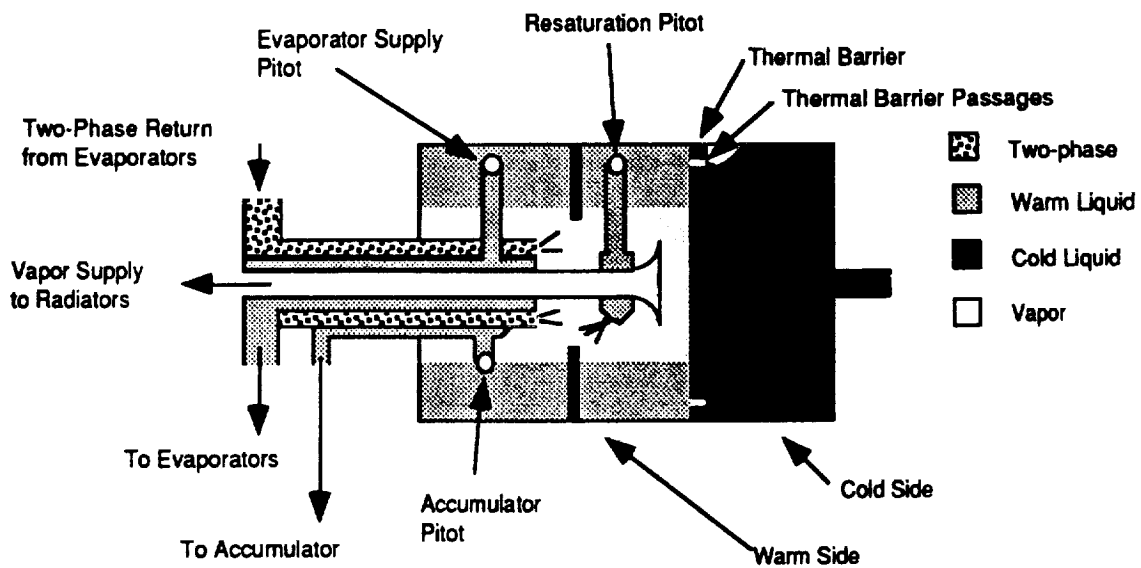


Figure 5. Rotary Fluid Management Device Schematic

Since the ETCS is a two-phase system, the operating pressure controls the operating temperature. The Back Pressure Regulating Valve is designed to passively control the pressure in the RFMD. The BPRV does this through a combination of spring and servo action (Figure 6). The system is also capable of changing operating temperatures by altering the relative spring forces inside the BPRV, thus temporarily affecting the heat rejection rate and raising or lowering the system pressure.

The modeling of the ETCS can be broken down into two distinct portions; the first being the modeling of the pump module assembly (RFMD, BPRV, and bellows accumulator) and radiator and the second being the remainder of the system (main plumbing, cavitating venturies, and module heat exchangers). The model of the pump module assembly consists of a detailed description of the RFMD, BPRV, and bellows accumulator. The model considers the physics of the RFMD by determining parameters such as the induced gravity head due to the rotation, pumping power from the pitot probes, and internal liquid levels. The model of the BPRV uses the internal geometry, including spring forces and internal bellows areas, to calculate the forces on the primary pressure control valve. The condensing radiator model is of moderate detail and accounts for thermal interactions between the parallel flow passages.

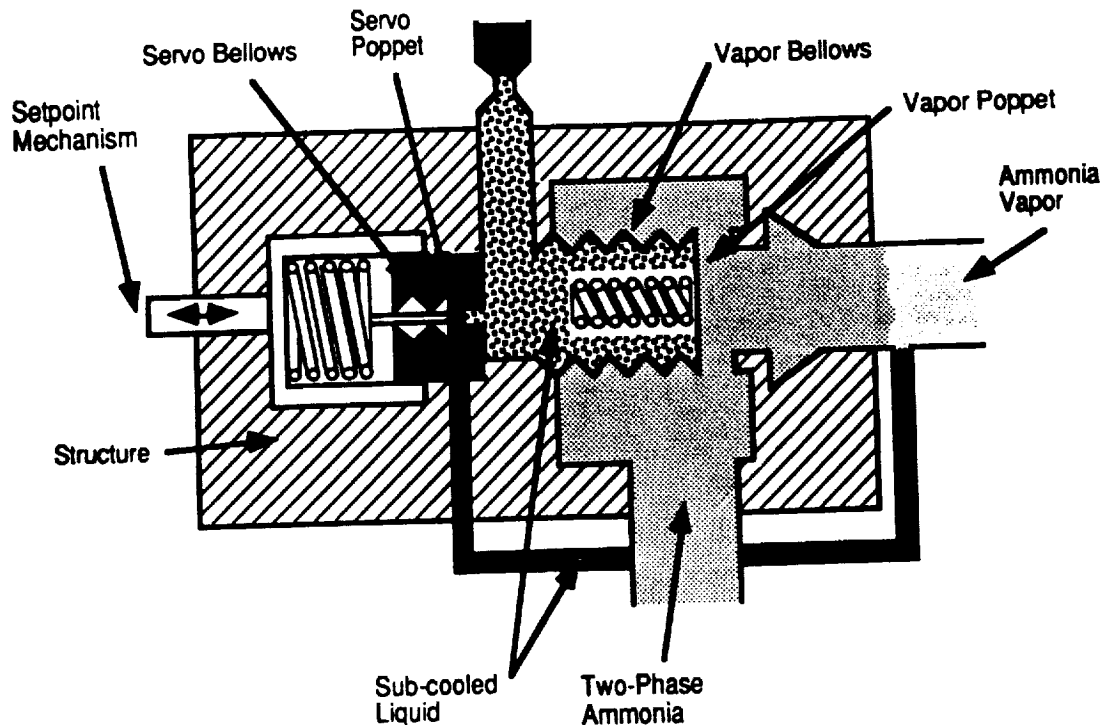


Figure 6. Back Pressure Regulating Valve Schematic

The model of the remainder of the system includes geometrically correct representations of the ETCS plumbing lengths, internal diameters, and plumbing routing. The correct geometry is critical to allow for accurate predictions of system pressure drops. The model takes advantage of the recently included option in FLUENT to determine pressure drops based on the flow regime.

Models of the cavitating venturies and module heat exchangers are also included. Due to the geometrically difficult phenomenon of the cavitating venturies, they are modeled utilizing a vendor supplied equation applicable for nominal conditions. Determination of cavitating venturi response during off-nominal conditions is still being investigated. The module heat exchangers are modeled as a single heat acquisition point since no details regarding the heat exchanger geometry are currently available. Figure 7 provides the FLUENT nodalization of the Moderate Temperature Loop for the Permanent Manned Capability configuration. The Low Temperature Loop's nodalization is similar.

Summary of Analyses Performed with the SSF Integrated ATCS Models

The models described above are used to analyze the Space Station active thermal control systems in both stand-alone and integrated modes. In stand-alone mode, the models are usually used to investigate control system dependencies, system pressure drops, and internal effects due to specific external conditions. These models have assisted in the detection and quantification of several system performance issues for both the United States and the International Partners.

For the United States modules, the models identified a mal-distribution of module waste heat onto the ETCS which could have caused an overload of the ETCS. Following a re-design of the U.S. modules, the FLUENT models assisted in control system development for the Resource Node. Also, the models were used to identify a potential water vapor condensation problem on a portion of the plumbing

that was un-insulated. The models were used to evaluate various options and to quantify the heat load value at which condensation became a potential problem.

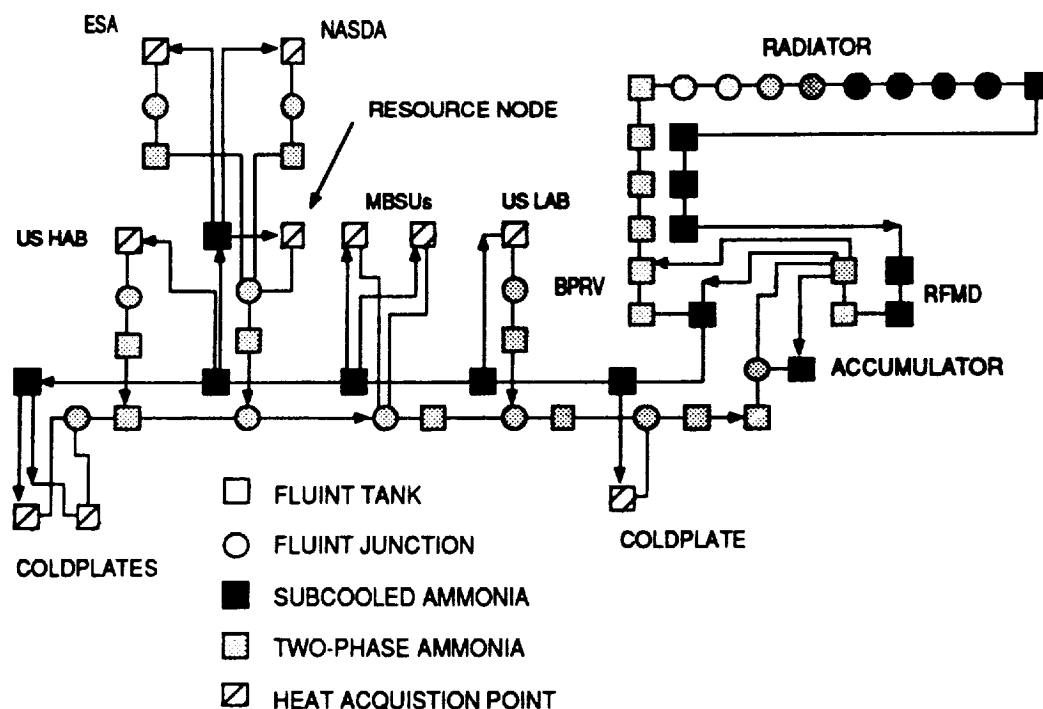


Figure 7. FLUINT Nodalization of the Moderate Temperature Loop for PMC

For the International Partners, the models identified a similar mal-distribution of module waste heat that violated Space Station operating specification. Several IP proposed options were modeled and evaluated as to the option's ability to meet station requirements. Additionally, the models have been used to evaluate control system stability.

In the integrated mode, the models have identified several key system issues and been used to analyze system operation at off-nominal conditions. One system issue identified was the inadequate heat load placed on one of the loops of the ETCS during early operations. The ETCS nominally consists of three loops - two Low Temperature and one Moderate Temperature. The Low Temperature loops are the first activated during the early stages of space station construction, when the total heat loads are the lowest. The pump module assembly requires a minimum amount of heat load (about 1.5 kW) to ensure adequate operation of the BPRV. Integrated analysis showed that only about 1 kW of thermal load was present on one of the Low Temperature loops during certain portions of station construction. The integrated models were used to devise methods to shift heat load from one Low Temperature loop to the other by altering the operating temperatures. Also identified by analysis in the integrated mode was the possibility of ammonia freezing in the condensing radiators due to low heat loads and cold external environments. The radiator manufacturer has confirmed this possibility and the design is being modified to account for freezing.

Off-nominal analysis has involved investigating the impacts to the system when operated at elevated temperatures. Operation at elevated temperatures enhances the heat rejection capability but can adversely impact the atmospheric temperatures inside the habitable modules. Since this change impacted the habitable modules, a coordinated analysis effort with the module providers was necessary. The ease with which the SINDA/FLUINT models can be changed allowed for rapid evaluation of proposed operating points.

The integrated models are currently being used to investigate the initial activation of the resource node, which is the first habitable module attached to the Space Station, and the impact of ETCS flow variations on both the ETCS and the internal modules. Other system responses being investigated are the transient response to operating temperature changes and the impact of large heat load changes, such as would occur during the re-activation of a module that was temporarily shut down.

DETAILED RADIATOR MODELS

Radiator Panel Description

Heat rejection in the Space Station ETCS occurs by condensation of ammonia in small diameter tubes of thermal radiators that radiate the heat to deep space. The radiators are designed to reject a total of 82.5 kW of waste heat. Of this, 49.0 kW is allocated to the moderate temperature (MT) loop and the remaining 33.5 kW is allocated to the two low temperature (LT) loops. The radiators are sized to satisfy heat rejection requirements of the LT radiators at a worst case design environment. This leads to a total heat rejection area of 4557 ft² (423.3 m²) for the LT loop radiators. The permanently manned capability (PMC) configuration of SSF has a total of 48 radiator panels grouped in 3 orbital replacement units (ORUs) of 8 panels, on each side of the Space Station, as shown in Figure 1. Each radiator panel consists of 22 thin-wall flow tubes manufactured of stainless-steel for ammonia compatibility. Each tube is 120.9" (3.070 m) long and has an inner diameter of 0.067" (1.7 mm). The flow tubes are inserted in a tube extrusion made of 6061-T6 aluminum alloy for high conductivity and low weight. Each tube extrusion is 0.67" (17.0 mm) tall with flat interfaces 0.232" (5.9 mm) wide on top and bottom. A very thin layer of thermal adhesive (0.003", 0.076 mm) is used between the stainless-steel flow tube and the aluminum extrusion to assure a good thermal contact between the two surfaces. The tube extrusions are bonded between two 6061-T6 aluminum face sheets 120.9" x 104.0" (307 cm x 264 cm) and 0.01" (0.254 mm) thick. The exposed surfaces of the face sheets are coated with a 0.005" (0.127 mm) thick film of Z93 paint for better radiative properties. Details of the flow tube and the extrusion are shown in Figure 8.

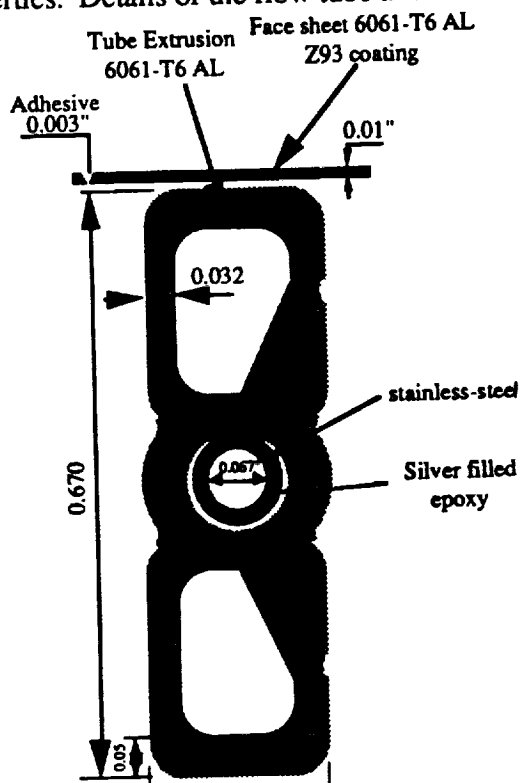


Figure 8 : Detail of the Tube Extrusion

Two separate sets of manifold tubes (manifolds A and B) carry vapor to and return condensate from each panel. Alternate tubes in each panel are connected to one set of manifold tubes. A diagram of the flow distribution to each panel in each ORU is shown in Figure 9. The manifold tubes at each end of the radiator panels are enclosed by manifold cover plates made of 6061-T6 aluminum and are 5.28" (13.4 cm) wide. The overall length of the panels including the manifold covers is 131.46" (333.9 cm). In order to provide support for the face sheets and increase the strength of the panels, the space between face sheets is filled with a layer of hexcel honeycomb material with a density of 3.1 lb/ft³ (49.6 kg/m³). The honeycomb is made of 5052 aluminum foil 0.0007" (0.018 mm) thick.

A moderate temperature (MT) ammonia loop (62° F, 16.6° C) flows through all port side radiator tubes. Two low temperature (LT) ammonia loops (35° F, 1.66° C) flow through alternating tubes in the starboard side radiators. No mass transfer occurs between the two LT ammonia loops, but they may communicate thermally by conduction through the face sheet.

The size of the plumbing of SSF has been optimized to minimize the weight while maintaining the pressure drop in lines at an acceptable level. Therefore, different size tubes are used in the LT and MT loops. Table 1 shows a summary of the plumbing sizes used in this study. The 3 ORU's at each side are pre-integrated on a section of the SSF truss in a folded position for easy transportation and can be deployed on orbit. This requires that the radiator panels in each ORU be connected by flexible tubing that can tolerate folding and unfolding. The manifold tubes of each panel in each ORU are connected to the

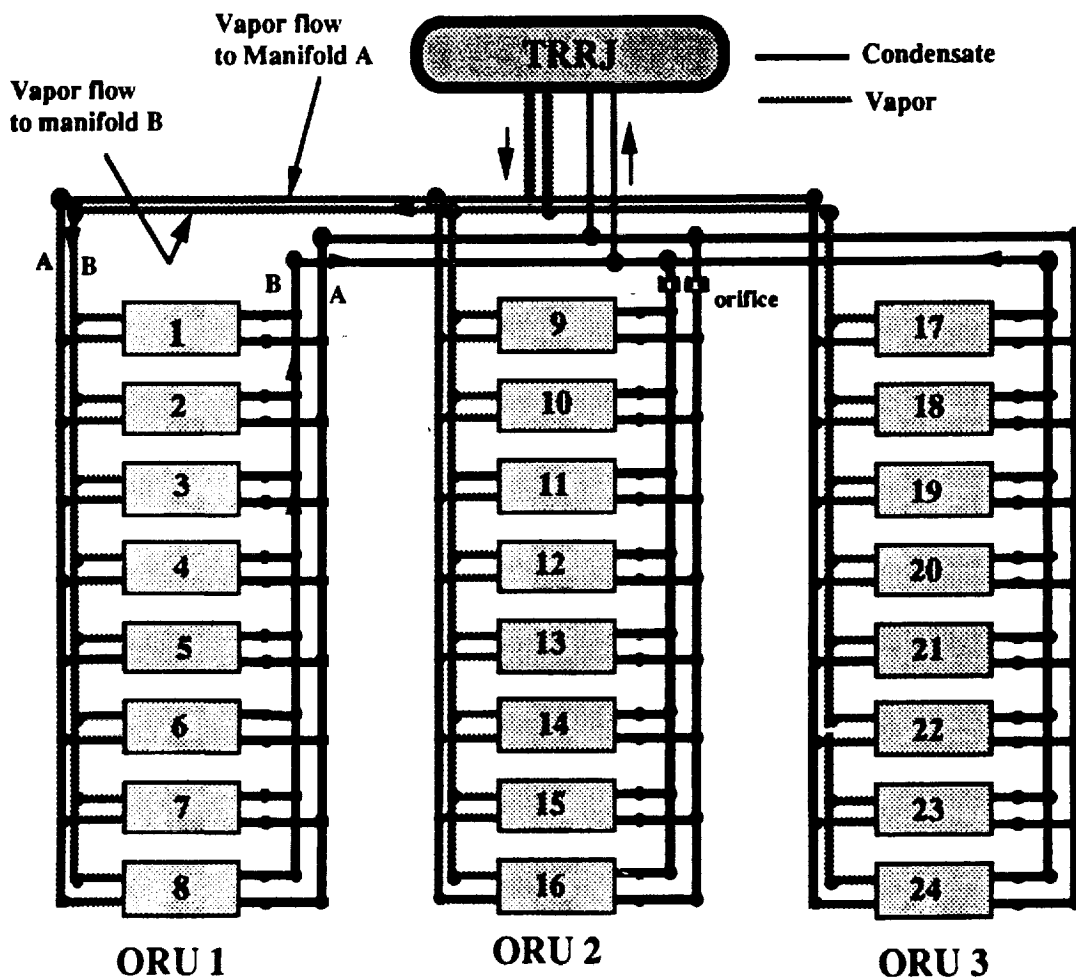


Figure 9 : Schematic of Manifolds and Radiator Panel Layout for one Wing

Table 1: Plumbing Size Summary

	35° F		62° F	
	PORT or STBD			
	ID (inch)	Length (ft)	ID (inch)	Length (ft)
MAIN TRUSS				
liquid	0.466	66.0	0.586	72.0
2-phase	0.952	66.0	1.20	72.0
PACK TRUSS				
liquid	0.466	72.0	0.585	116.0
2-phase	0.706	72.0	0.952	116.0
UMBILICAL				
liquid	0.346	20.0	0.346	20.0
2-phase	0.586	20.0	0.586	20.0
CONDENSER LOOP				
cond. to RFMD	0.466	50.0	0.466	50.0
vapor to panels	0.952	50.0	0.952	50.0
CONDENSERS				
manifold tube	0.939	8.0	0.939	8.0
flex hose	0.805	2.0	0.805	2.0
panel flow tube	0.067	10.07	0.067	10.07

manifold tubes of the next panel in line by flex hose assemblies as shown in Figure 10. The flex hose assembly consists of two pieces of flex hoses and a 180° bend tube section. The loss factor of the flex hoses is estimated as 1.4/ft.

Description of the Detailed Radiator Models

The models described here were developed for detail analysis of the heat rejection system of the ETCS. A simplified representation of the other ETCS subsystems was made in order to maintain size of the models and save computer processing time. The models were developed on version 2.4 of SINDA/FLUINT (ref. 1). Several models with various levels of detail were developed for each configuration of SSF, for MT and LT radiators. However, this study is focused on the analyses performed on the LT radiator models.

Several assumptions were made in developing the models used in the analysis of LT loop radiators. The assumptions were made based on available information at the time the analyses were conducted and the attempt was made to justify validity of these assumptions by comparing the results of the analysis with available data. The water/ammonia interface heat exchangers (IHX) and coldplates (CP) were modeled as point heat sources. The flow rate to each evaporator was calculated by the model such that at the maximum heat load the evaporator vapor quality was 0.8 for the heat exchangers and 0.9 for the coldplates. The estimated pressure drop associated with each IHX or CP was modeled by a LOSS element in line with that IHX or CP.

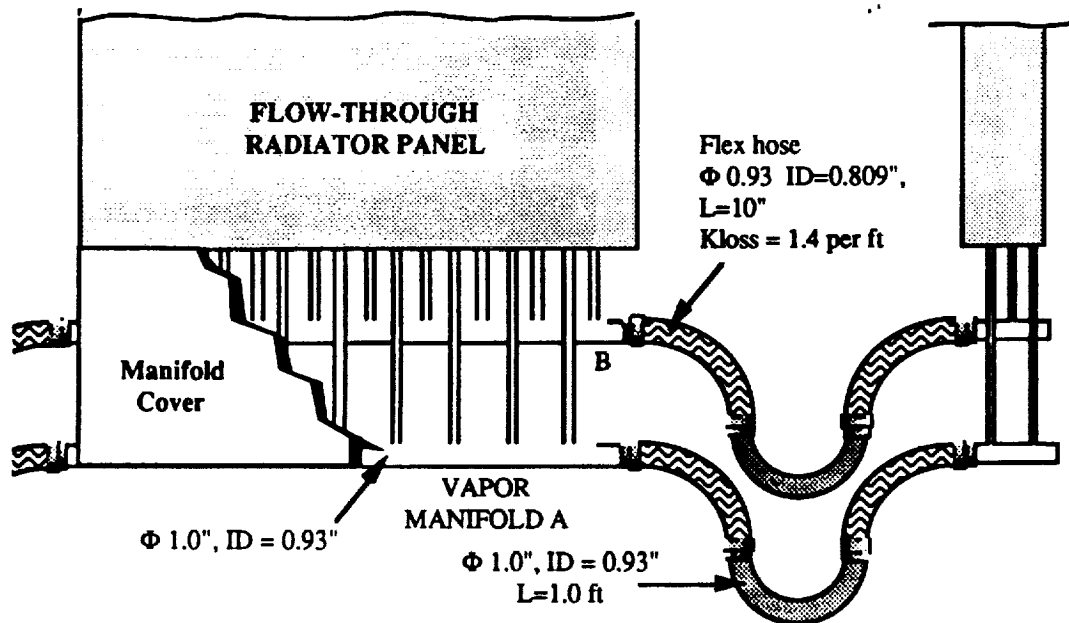


Figure 10 : Arrangement of the Manifold Tubes and Flex Hoses

The rotary fluid management device (RFMD) was modeled by two plenum - one representing the warm end set at saturation pressure corresponding to the desired set point temperature, and the other representing the cold end set at subcooled condition. A pressure difference equal to the RFMD end-to-end pressure difference (as reported by the manufacturer) was imposed on two plenum. A saturated liquid flow rate equal to the evaporator flow rate plus the RFMD bearing flow and the back pressure regulating valve (BPRV) servo flow was extracted from the warm end plenum. The vapor flow rate out of the RFMD was calculated dynamically by performing an energy and flow balance on the RFMD. The vapor flow rate is a function of the instantaneous heat load, set point of the RFMD, evaporator and condenser return flow conditions, and bearing and servo flow rates. The vapor flow mixes with the BPRV servo flow before entering the radiator panels. The servo flow rate was assumed to vary linearly with total heat load on the system from 0.022 GPM at maximum heat load to 0.015 GPM at minimum heat load (3.35 kW). The BPRV is designed to maintain the RFMD warm end pressure at the set point saturation pressure by regulating the vapor flow rate. The BPRV opening (which determines BPRV pressure drop) is then adjusted to allow sufficient vapor flow out of the RFMD. The model of the BPRV performs the same functions for the normal operations of the system. The BPRV pressure drop in the model varies with vapor flow rate in order to balance the RFMD end-to-end pressure drop. The model will signal if the BPRV reaches its maximum opening and can no longer control the set point.

Several versions of the radiator model with different levels of detail were developed for different analyses. The alternating flow tubes in each radiator panel connect either to LT loop A or B. The flow tubes and other lines and components corresponding to each loop were represented in a separate submodel since no flow mixing occurs between the two loops. Each panel was then represented by models of one flow tube from loop A and one from loop B which were thermally connected through conduction in the face sheet nodes. Each flow tube was modeled by a HX macro with 10 segments in the direction of the flow. The ammonia in each section was represented by a JUNC and the flow tube section was represented by a TUBE with each lump downstream of each tube section.

Condensation Heat Transfer Correlation

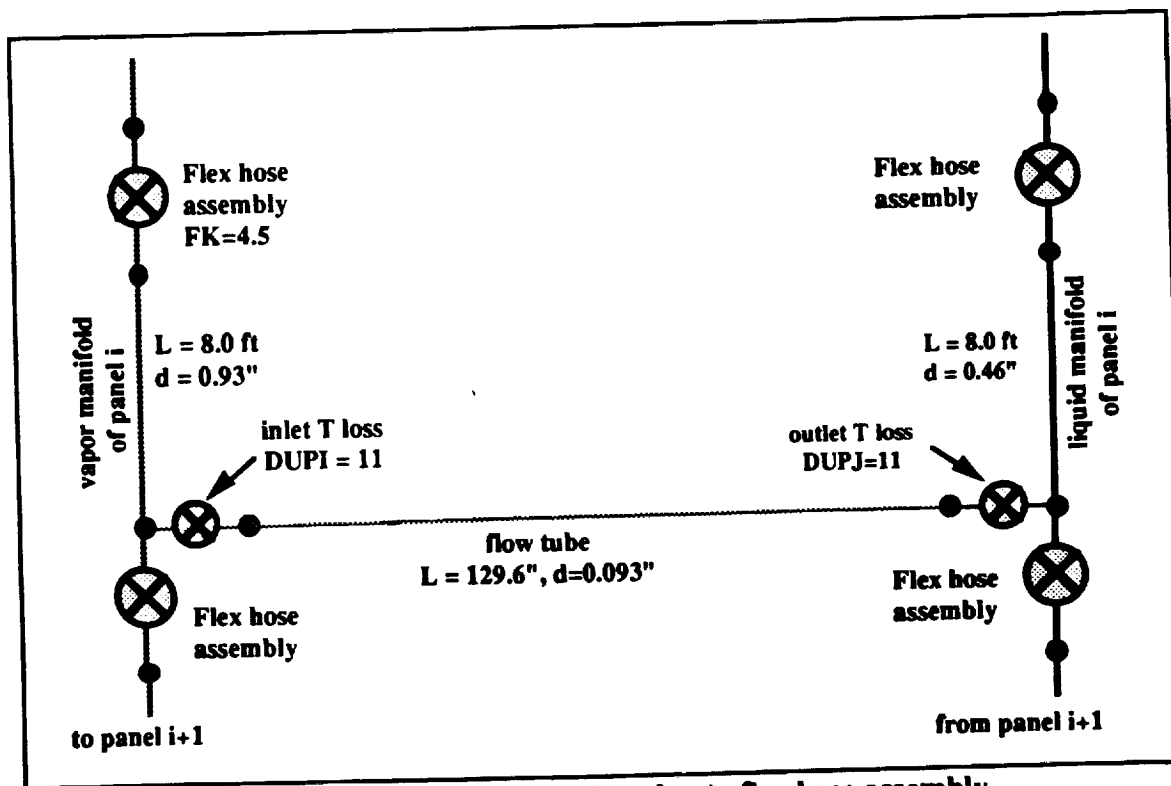
The condensation heat transfer coefficient in FLUINT is based on Rohsenow's correlation (ref. 2). This correlation was developed for condensation in annular flows and uses the Martinelli parameter in calculating the two-phase Reynold's number. Transition to single-phase liquid occurs at qualities less than 0.10 and accommodates the breakdown of annular flow into slug flow, using scaling parameters similar to Shah's correlation (ref. 3).

Pressure Drop Correlation

Generally speaking, no single pressure drop correlation can produce accurate predictions of pressure drop for all fluids and flow conditions. The prediction of pressure drop for micro gravity conditions is even more difficult due to uncertainty about the exact flow regime. FLUINT offers a homogeneous (default) and several two-phase pressure drop correlations by setting the IPDC (pressure drop correlation selector) from 0 to 6. Based on the available information at the time of this study, the Lockhart-Martinelli correlation (IPDC=2) was found to best approximate the available data and was used in the models.

Manifold tubes and flex hoses

The schematic of the manifold tubes and flex hoses model for one panel is shown in Figure 11. The loss coefficient and length of the flex hoses that connect the manifold tubes of the adjacent radiator panels were given in Figure 10. A loss coefficient of 4.5 was used for the flex hose assembly which was modeled by a LOSS connector. Additional LOSS connectors were used to model the pressure drop at the vapor manifold-to-flow tube tee and flow tube-to-condensate manifold tee.



Representation of pressure loss due to flex hose assembly and T's in flow-through panel model

Figure 11: Manifold and Flex Hose Model

Thermal Network

The radiator panel face sheets and extrusion tubes were divided into 10 strips along the panel length. The face sheet area corresponding to each tube segment was 4.73" (12.0 cm) wide, 12.09" (30.7 cm) long, and centered above the tube. The distance of 4.73" (12.0 cm) on the face sheet between the two adjacent tubes modeled was divided into 9 nodes. This nodal breakdown allowed for determination of the temperature profile of the face sheet between two tubes and eliminated the need for a pre-determined fin efficiency for the radiator face sheet. The nodal breakdown of the radiator face sheet is shown in Figure 12. The mass of each tube segment was represented in the capacitance of the tube wall nodes which was thermally tied to the fluid lump in that tube segment via a heat transfer tie. The flow tube inner wall node was connected to the extrusion inner surface via conductance through the stainless-steel and tube adhesive. The overall conductance of the extrusion tube, from its inner surface to its interface surface, was calculated from a separate detail SINDA model of the extrusion tube. The extrusion interface node was connected to the face sheet node via conduction through the adhesive. The extrusion tube nodes were also connected axially to allow for axial conduction in the subcooled part of the tube. The face sheet nodes were connected both laterally and axially. The thermal network of one segment of two adjacent tubes is shown in Figure 13.

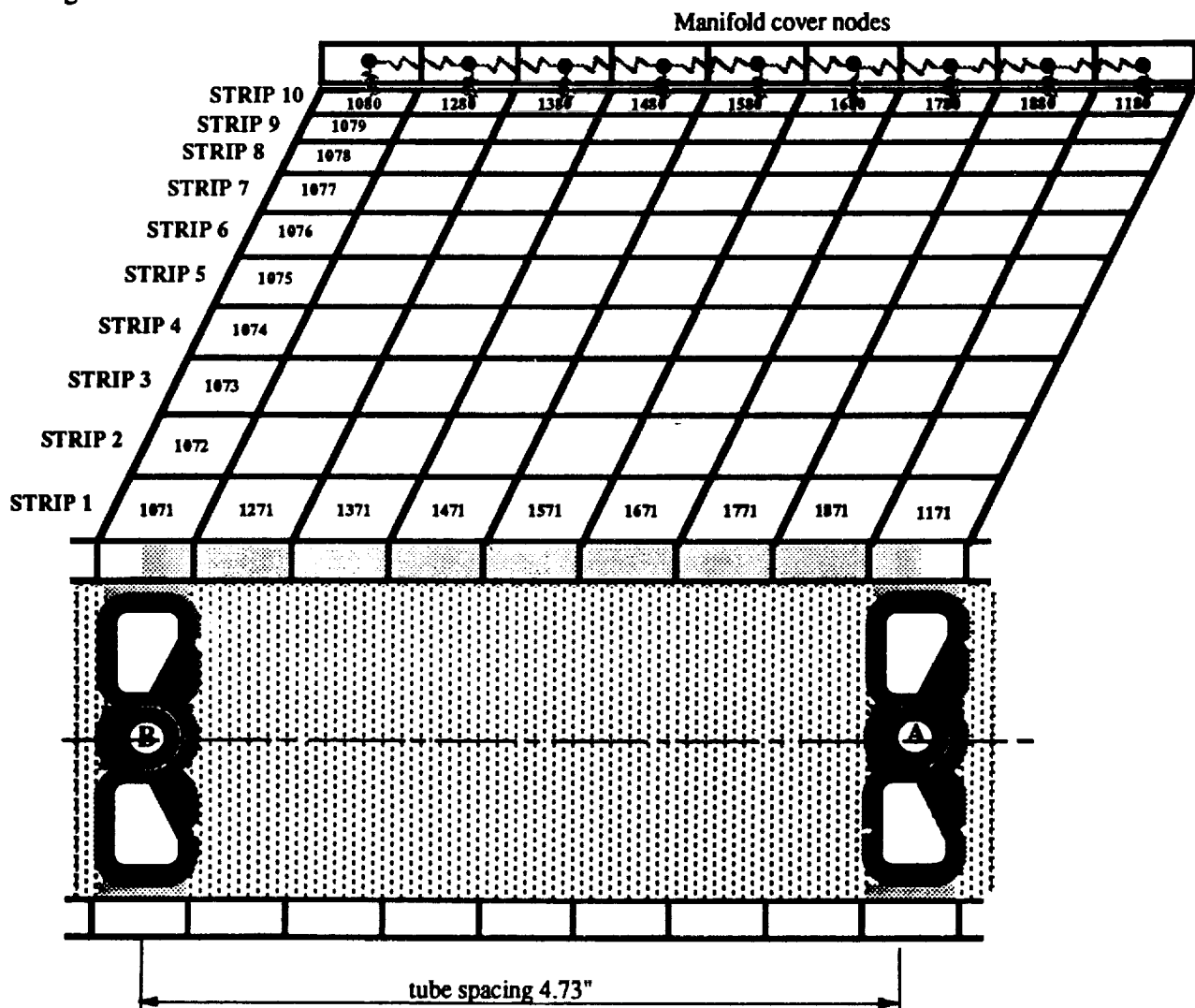


Figure 12: Nodal Breakdown of Radiator Facesheet

The model of the radiator panels was built by using the above described techniques and using a series of DUP's. The DUP option in FLUINT allows the user to simulate the presence of several identical paths without actually modeling them. Each flow tube was DUPed 11 times to represent the 11 tubes per loop in each panel. Each panel was then DUPed 8 times to represent one radiator ORU, and the ORU was in turn DUPed 3 times for representation of the entire radiator array. Another model of the radiator system was developed in which all 22 tubes in one panel were actually modeled. This model was used to study flow variations among the tubes in each panel. A model was also developed in which each panel was represented by two adjacent tubes DUPed 11 times, but all 24 panels were actually modeled. This model was used for study of flow variations among radiator panels due to frictional pressure drop, effects of different environments on each radiator, and effects of isolating some panels or ORU's.

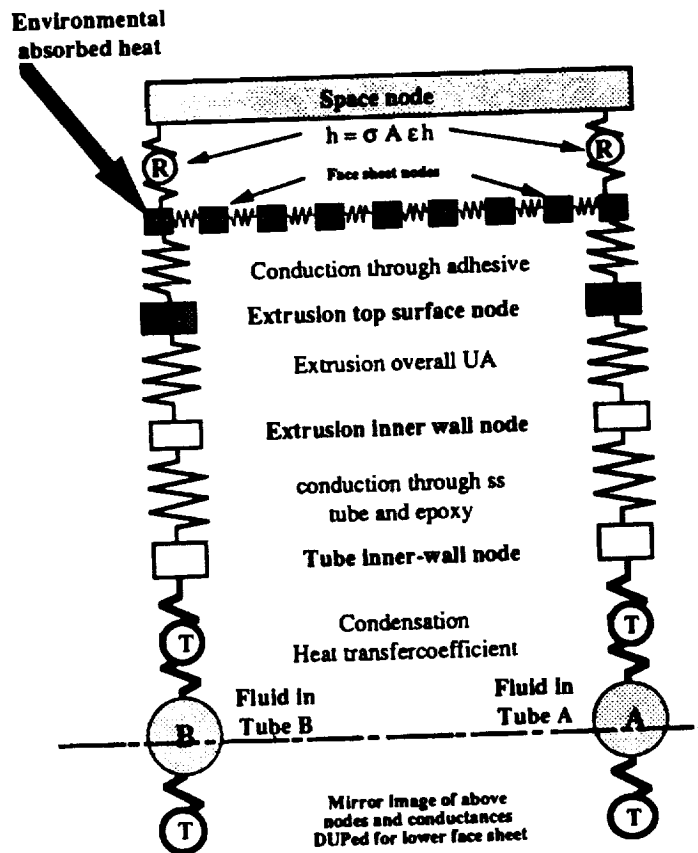


Figure 13: Nodal Breakdown of Radiator Flow Tubes

Counter-flow Radiator Model

The baseline design of the ETCS radiators (described above) consists of two separate loops that flow in alternating tubes of the radiators in the same direction. As a means of increasing the freeze tolerance of the radiators, an alternative arrangement called counter-flow was considered in which the flow in every other tube was in the opposite direction. The arrangement of the manifolds for the parallel (baseline) and the counter-flow option are shown in Figure 14. In the counter-flow option, conduction between loops through the face sheet is very important since the cold end (outlet) of each flow tube is heated by the warm two-phase end of the adjacent tube(s). Results from this study will be presented in addition to the results for the baseline design.

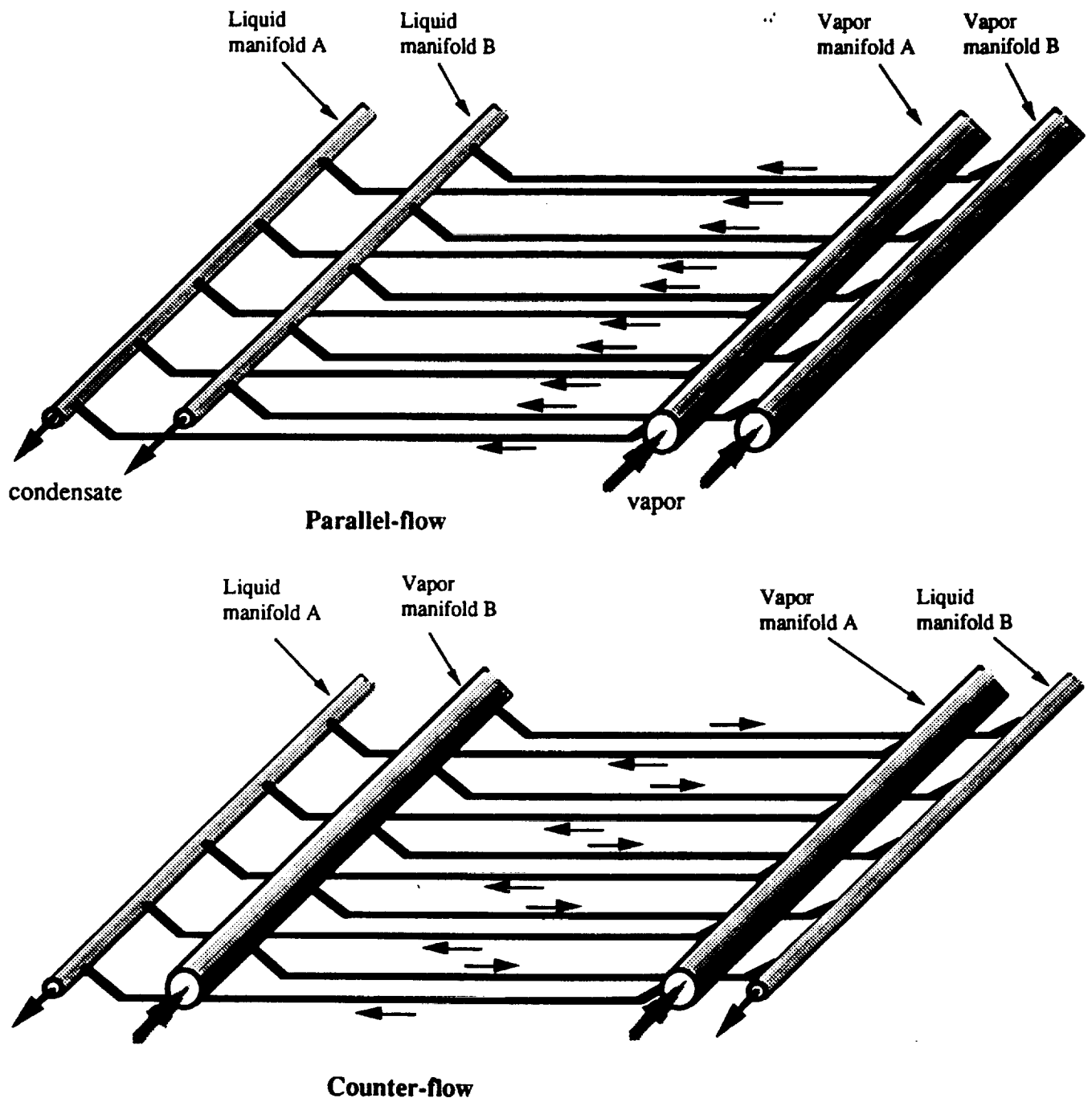


Figure 14: Schematic of the Radiator Panel Manifolds for Parallel and Counterflow

Detailed Radiator Model Analysis Results

The models were analyzed at steady-state and transient conditions for a variety of cases as described below. The Space Station configuration analyzed was the Man Tended Capability (MTC) stage with only one of the radiator ORUs deployed.

MTC Configuration, Warm Case

The MTC configuration model was analyzed at 15 kW heat load and an equivalent effective sink temperature of -50°F (-45.5°C). The heat load was equally split between loops A and B. The results for the two loops were identical and are shown in Table 2. The results indicated that 2-phase ammonia with a vapor quality of 0.967 at 29.3°F (-1.5°C) enters the radiators and exits as subcooled liquid at -32.7°F (-35.9°C). 72.6% of the length of each flow tube was condensing and the rest was subcooling the flow. The panel overall fin efficiency under these conditions was 0.762. The panel overall fin efficiency is the ratio of panel actual heat rejection to ideal heat rejection if the temperature of all the face sheet nodes were equal to the temperature of the hottest face sheet node.

Table 2 : Results of the MTC LT loop analysis at -50°F sink temperature

	Flow Rate Lb/hr	Heat Load [KW]	Inlet Temp [F]	Inlet quality	Outlet Temp [F]	Condensing Fraction
Loop A	0.498	7.5	29.37	0.967	-32.7	0.726
Loop B	0.498	7.5	29.37	0.967	-32.9	0.726

MTC Configuration, Cold Case

An effective sink temperatures of -95°F (-70.5°C) was used to represent a typical radiator cold environment (no freezing). The results of the MTC configuration analysis for a 15 kW total heat load are shown in Table 3. The results indicated that the ammonia outlet temperature was subcooled at -83.9°F (-64.4°C).

The model was run for the same cold case conditions as above with a radiator counter-flow configuration to examine the effectiveness of the counterflow design in raising the radiator outlet temperature. The results shown in Table 3 indicated that the radiator outlet temperature was 84.5°F (47°C) warmer for the counter-flow case. The ammonia temperature profiles in each flow tube for the parallel and counter-flow cases are shown in Figure 15. The results indicated that although condensation was completed faster in the counter-flow case, the subcooling of the condensate flow at the outlet was much less due to heat leak from the 2-phase ammonia in the adjacent tube. The coldest ammonia temperature in the counter-flow case was -60°F (-51.1°C) and occurred in the middle of the panel. The panel overall fin efficiencies for the parallel and counter-flow cases were 0.54 and 0.55, respectively.

Table 3 : Results of the MTC LT loop analysis at -95°F sink temperature

	Flow Rate Lb/hr	Heat Load [KW]	Inlet Temp [F]	Inlet quality	Outlet Temp [F]	Condensing Fraction
Parallel flow	0.457	7.5	29.1	0.97	-83.9	0.48
Counter- flow	0.528	7.5	29.1	0.97	0.6	0.29

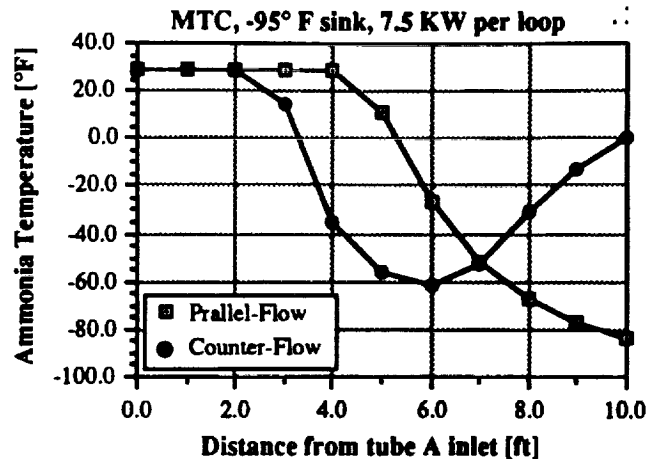


Figure 15: Ammonia Temperature Profile in Flow Tube A for Parallel and Counterflow

Radiator Freeze Prevention Studies

Under low heat loads and very cold environments when the effective sink temperature falls below the freezing temperature of ammonia (-108°F, -78°C), the fluid in the radiators may freeze. This leads to very high local pressures inside the flow tubes when frozen ammonia begin to thaw. During the SSF ETCS design process, the problem of radiator freezing (and subsequent thawing) was addressed. One approach was to try to increase the radiator condensate return temperature in cold cases without limiting its heat rejection capability in hot cases. Other approaches allowed the ammonia to freeze and thaw under controlled conditions. Several of the options which were investigated using the detailed SINDA/FLUINT radiator models are presented here.

Freeze Prevention by Radiator Pointing

The MTC configuration of SSF was analyzed with two heat loads at transient conditions using a cold case environment (orbit angle Beta=0°, beginning of life surface properties) and different radiator orientations. The results are shown in Table 4. The coldest environment was the "edge-to-sun on the sun side"/"edge-to-earth on the dark side" (ETS/ETE) orientation, which resulted in radiator freezing for some

Table 4 : Summary of the results of MTC configuration transient analysis under cold environment at $\beta=0$, for different radiator panels orientation

PANEL ORIENTATION sunside/darkside	NO. of Panels	Condensate return temp. [° F]	
		4.0 kw	11.2 kw
ETS/ETE	8	freezing	freezing
ETS/FTE	8	-106	-95 to -100
FTE/FTE	8	-29 to -51	-
45°-to-earth	8	-62 to -75	-33 to -81

parts of the orbit for both heat loads. The "face-to-earth on the sun side"/"face-to-earth on the dark side" (FTE/FTE) orientation was the warmest environment, but resulted in a reduced heat rejection capacity of the radiators on the sun side of the orbit. The results indicated that the situation was improved and freezing was prevented as panel orientation was changed to face the earth on the dark side of each orbit. However, the large pitch angle in the SSF torque equilibrium attitude (TEA) significantly limited the possibility of radiator pointing to modifying the radiator environment, and resulted in environments that were much colder. Therefore, this option alone was not sufficient to prevent radiator freezing.

Freeze Prevention by Increasing BPRV Liquid Flow or Heating Radiators

Using the SINDA/FLUINT models, a study was conducted to examine the feasibility of other options such as increasing the BPRV liquid servo flow or electrically heating radiator flow tubes in order to prevent radiator freezing (ref. 4). The 4-orbit cold case design sink temperature profile for $\beta=0^\circ$ and $TEA=0^\circ$, shown in Figure 16, was used in the study. The results of the study showed that increasing BPRV servo flow had no significant effects on radiator outlet temperature. The change in radiator outlet temperature was negligible even when BPRV servo flow rate was increased by 20 times its normal value. Next, the model was analyzed at 3.35 kW heat load with simulated electrical heaters bonded to the extrusion tubes. The heaters were turned on only during the dark side of each orbit. The results indicated that 5.28 kW of electrical power was required in order to maintain radiator outlet temperatures above -90°F (-67.8°C). Use of that much power would have an unacceptable impact on SSF.

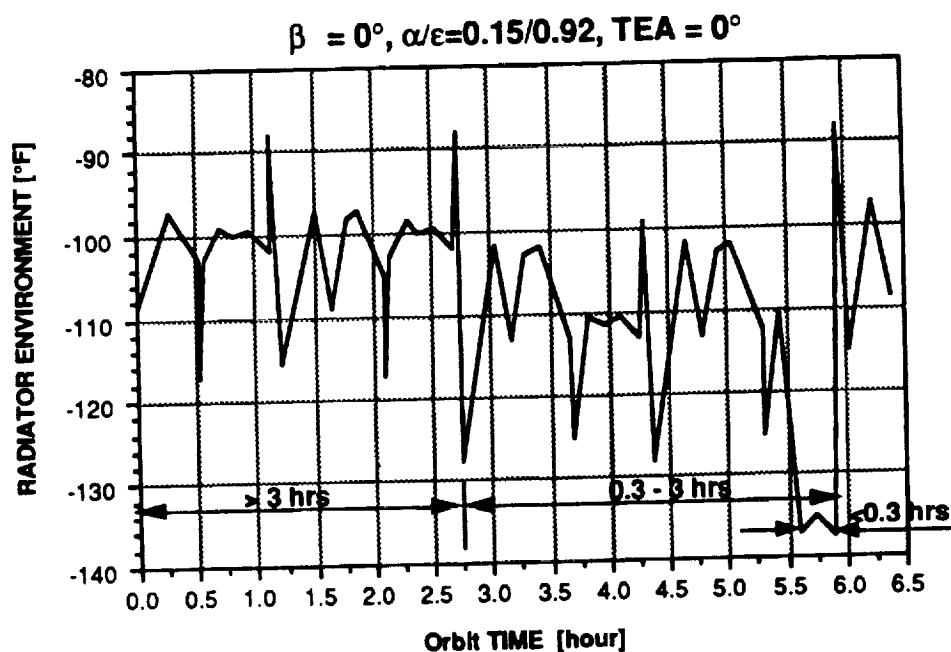


Figure 16: Cold Case Sink Temperature Profile for Coldest Panel

Freeze Prevention with Counter-flow Radiators

The counter-flow radiator design (described above) was another option that was considered since it does not increase the weight of the radiators and does not require any additional components. This option eventually gave way to a SSF design solution in which some of the condensate tubes were allowed to freeze and thaw without bursting. For the SSF case, the freeze/thaw design solution was used since it works even when one of the low temperature loops is not flowing; however, in other applications the counter-flow option may be the best design solution to deal with radiator freezing.

A case was run using the MTC model at low loads (3.35 kW per loop), using the coldest panel environment for $\beta=0$ and $TEA=-45^\circ$ as shown in Figure 17. The results of the analysis using the parallel-flow (baseline) option are shown in Figure 18. The results showed that freezing of ammonia occurred at sub-freezing sink temperature parts of the orbit. The results of the analysis with the counter-flow model are shown in Figure 19. Unlike the parallel flow case, the coldest fluid temperature in the counter-flow design occurs at the middle of the radiator panels since the condensate return flow at the outlet of each tube is warmed up by the 2-phase flow at the inlet of the adjacent tubes. Figure 19 shows the radiator condensate return temperature as well as the radiator coldest fluid temperature and indicates that the freezing problem was eliminated by the counter-flow option.

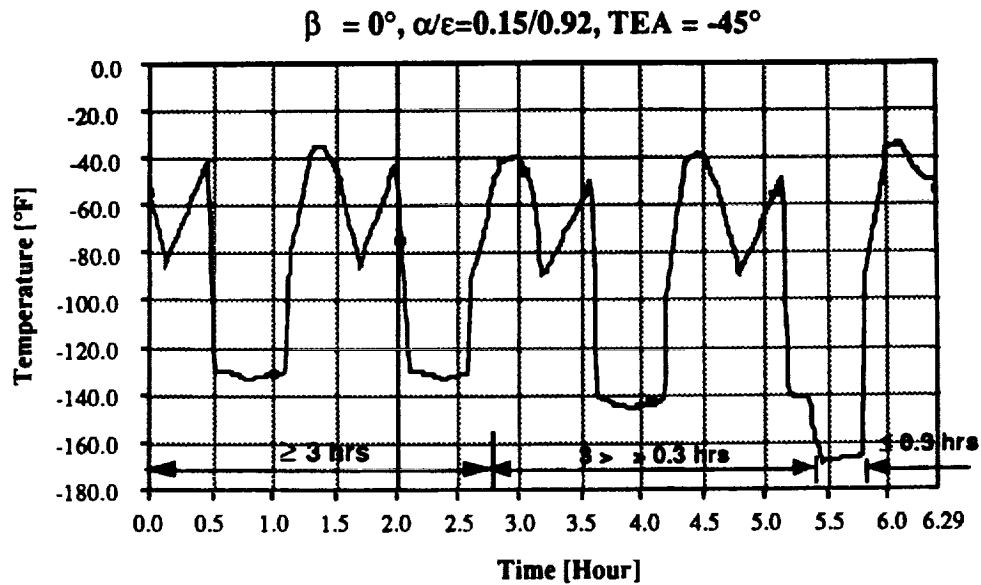


Figure 17: Coldest panel environment for MB5 configuration

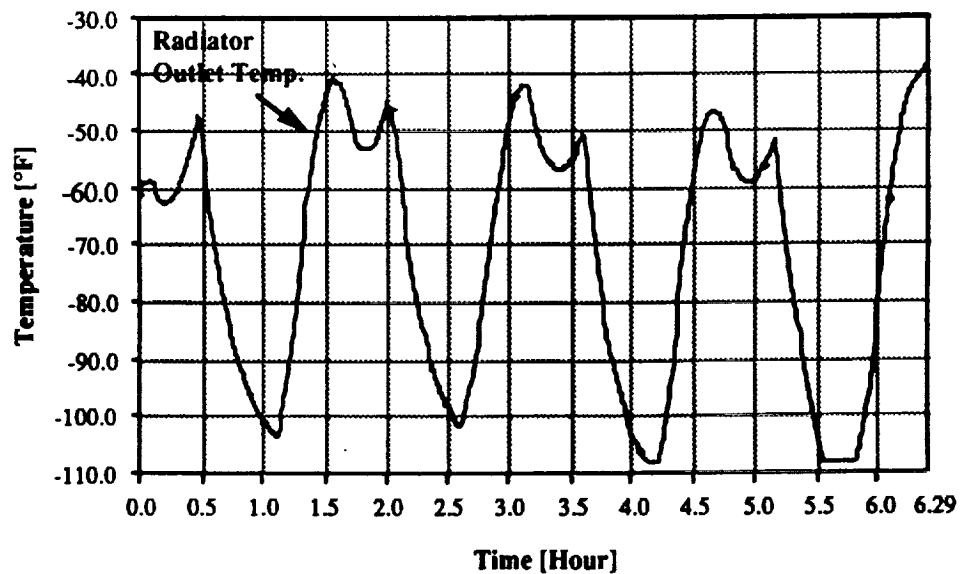


Figure 18: Parallel-flow Radiator Exit Temp. ($\beta=0^\circ$, $TEA=-45^\circ$, 3.35 kW per loop)

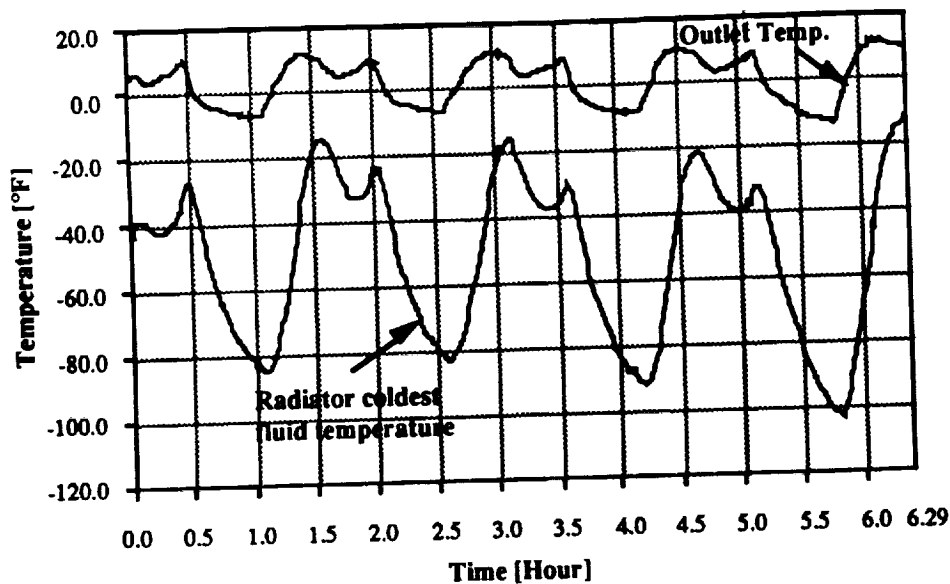


Figure 19: Counterflow Radiator Temperatures (Beta=0°, TEA=-45°, 3.35 kW per loop)

Detailed Radiator Models Summary

SINDA/FLUINT is a powerful tool for detailed analysis of complicated thermal and fluid systems. The detailed models developed at JSC for the analysis of the Space Station ETCS radiators have been essential in predicting performance of the radiators for many nominal and off-nominal conditions of practical interest. The models have been used to study such design problems as response to orbital environment transients and radiator freezing.

IDEAL GAS SYSTEMS

The Space Shuttle's ATCS, the SSF Internal Thermal Control System, and the flow of propellant are all aerospace systems involving single-phase liquid flow. In addition to these applications, the filling and evacuation of rigid containers by a gas is another situation where single-phase flows are encountered.

From thermodynamics, when a gas is compressed or expanded isentropically and adiabatically, its temperature will increase or decrease, as described by the integrated, isentropic forms of the Gibb's equation (decompression) or the first law of thermodynamics (compression) modified by the ideal gas law [eqn. 1]. However, if this gas is contained in a non-adiabatic tank, heat transfer with the side walls will affect the thermal response of the gas and the simple equations relating pressure and temperature will be complicated. In general, the introduction of heat transfer terms into these governing equations prevents a closed-form solution and a numerical model must be developed. The SINDA/FLUINT program can be used to solve these problems.

The Basic SINDA/FLUINT Model

Figure 20 presents the schematic of a basic SINDA/FLUINT model that can be used to predict pressurization and depressurization of a gaseous system. The gas storage container is represented by the TANK option which allows calculation of transient pressure and temperature changes as mass is removed from or added to the system. To account for possible environmental heat losses, the TANK is tied thermally to a SINDA model (or an individual node) of the storage container by the use of a convection

heat transfer (HTU) tie. The value of this tie must be determined for each specific application. System flow rates are controlled by the MFRSET (Mass Flow Rate SET). The plenum (PLEN) is used to represent an infinite source of supply gas or a reservoir to dump the exhaust gas. The TANK is connected to the PLEN by an STUBE (Short TUBE) which can, if User Logic is employed, represent a pressure drop device (such as a regulator or an orifice). The working fluid is modeled as an ideal gas by using 8000 series fluid data blocks.

To ensure that this model was properly developed, the predicted final temperatures were compared against the theoretical results for adiabatic filling and isentropic, emptying processes for dry air. For a tank emptying process with constant thermophysical properties, the equation for the final temperature is (ref. 5),

$$T_f = \frac{T_i}{\left(\frac{P_i}{P_f}\right)^{\frac{k-1}{k}}} \quad [1]$$

where T_f is the final temperature, T_i is the initial temperature, P_f is the final pressure, P_i is the initial pressure and k is the ratio of the specific heats.

For a filling process, the following equation may be used (ref. 6),

$$T_f = \frac{kP_f T_{in} T_i}{(P_f - P_i)T_i + kP_i T_{in}} \quad [2]$$

where T_{in} is the temperature of the inlet gas.

For the range of parameters to be considered here, the predicted results from the SINDA/FLUINT model were within 0.5 °F (0.3 K) of the theoretical results of both equations. The slight discrepancy is due to use of constant properties (k) in the theoretical equations, while the numerical model employs variable properties. From this comparison study, it was felt that the model was properly developed for filling and evacuation processes.

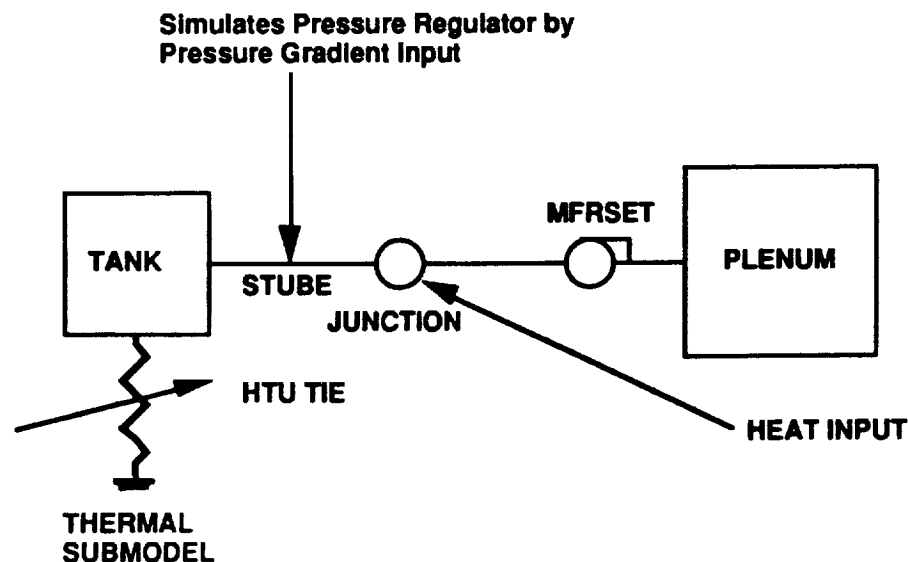


Figure 20: Schematic of the Gaseous System SINDA/FLUINT Model

Space Station Freedom Airlock

During the life of the Space Station Freedom (SSF), the crews will be required to perform a variety of Extra-Vehicular Activities (EVAs), in order to maintain the station. Before entering the vacuum of space, the EVA teams will perform a variety of breathing exercises to condition themselves for the EVA. Once these breathing exercises are completed, the crew enters the airlock, evacuates the chamber by using a vacuum pump and journeys outside.

If during the EVA (or some other time) a crew member is exposed to a low pressure environment, it is possible that he will experience decompression sickness (the bends). To avoid returning to Earth and to provide quick medical treatment, the SSF airlock will be used as a hyperbaric treatment facility. To perform this medical treatment process, the airlock is pressurized to 2.8 atmospheres and the affected crew member(s) undergo a variety of breathing exercises. To reduce the high temperatures associated with compression, a heat exchanger is included as part of the airlock hardware.

As evident from equations [1] and [2], the gases within the airlock will exhibit substantial changes in temperature. In addition to fluid temperature variations due to pressure changes, heat transfer to the side walls and heat removal by the heat exchanger will also affect system response. To understand how these phenomena interact and affect the airlock gas temperature and pressure, a detailed model including all the aforementioned effects was developed.

Development of the Numerical Model

Figure 21 shows a schematic of the airlock which is depicted as a right circular cylinder, 6.22 ft (2.00 meters) in diameter and 8.22 ft (2.50 meters) in length. Located on one wall of the airlock is a fan and cross-flow heat exchanger assembly which is used to cool the chamber during hyperbaric operations. Also included in this package is a centrifugal vacuum pump which is used to depressurize the chamber for EVAs.

Before a numerical model of this system could be developed, certain simplifying assumptions for both the air and specific mechanical aspects of the airlock were made. These assumptions are listed below.

- 1) The gases within the airlock may be considered ideal. For the conditions examined here, the compressibility factor (Z in most thermodynamics textbooks (ref. 6)) is nearly unity, indicating ideal gas conditions. As a result, the thermophysical properties of gas considered varies only with temperature.
- 2) At any given time, the gas within the chamber is at a uniform temperature. That is, there are no temperature gradients across the airlock gas.
- 3) Since the flow inside the airlock during the withdrawal process is low velocity and laminar, convection effects will be ignored. On the other hand, during hyperbaric operations the fan is on and produces substantial flow velocities. To account for this convection, the heat transfer coefficient will be the same as that of the Space Shuttle.
- 4) Due to the lack of gravity, there are no natural convection effects.
- 5) The gas is dry so heat transfer effects due to condensation are ignored.
- 6) The gas within the airlock is not a participating media.
- 7) The airlock is constructed out of 6061 aluminum and at a uniform temperature.
- 8) The heat exchanger is modeled using the NTU method (ref. 7).
- 9) There are no heat losses to the environment.

Figure 22 shows a schematic of the SINDA/FLUINT airlock model, which is the same as the model shown in Figure 20 with several minor modifications. The gas is represented with the TANK option and is tied to a single thermal diffusion (time-dependent) node which represents the airlock's metal mass so that convective heat transfer effects can be included. The PLEN represents either the hyperbaric

charging tanks or the vacuum of space. The MFRSET has been replaced by a VFRSET (Volume Flow Rate SET) to provide a more accurate representation of the vacuum pump.

The hyperbaric heat exchanger is modeled with its own loop, and more specifically with a pair of junctions. To determine its heat removal rate, the following approach was taken. First, the NTU method (ref. 7,8) was applied using the characteristics of a preliminary heat exchanger design (ref. 9) and its outlet air temperature was determined. With the outlet thermodynamic state determined, the CHGLMP (CHanGe LuMP) option was used to alter the current state to the new and more accurate condition and the HTRLMP (HeaTeR LuMP) option was used to hold the current state. The HTRLMP option maintains the desired thermodynamic state by supplying an appropriate heat load at the downstream junction.

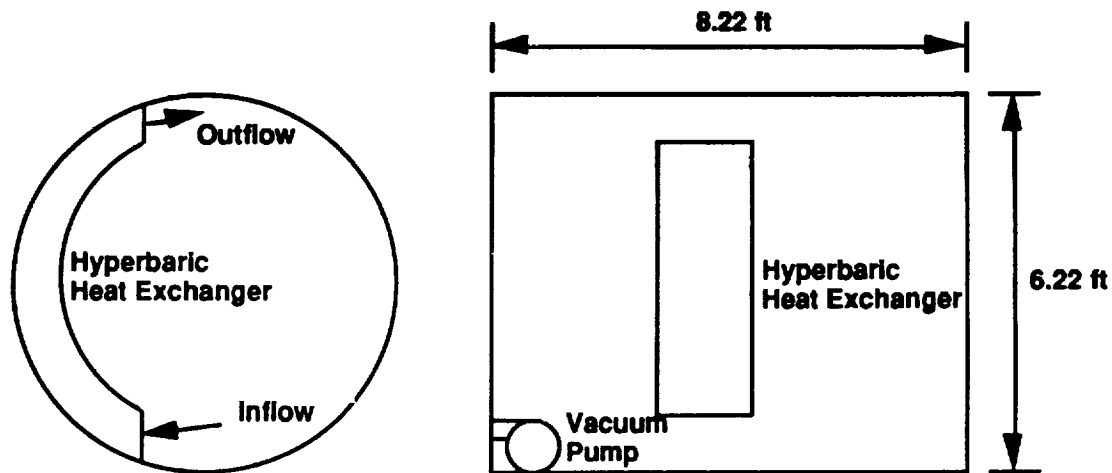


Figure 21: Schematic of SSF Airlock.

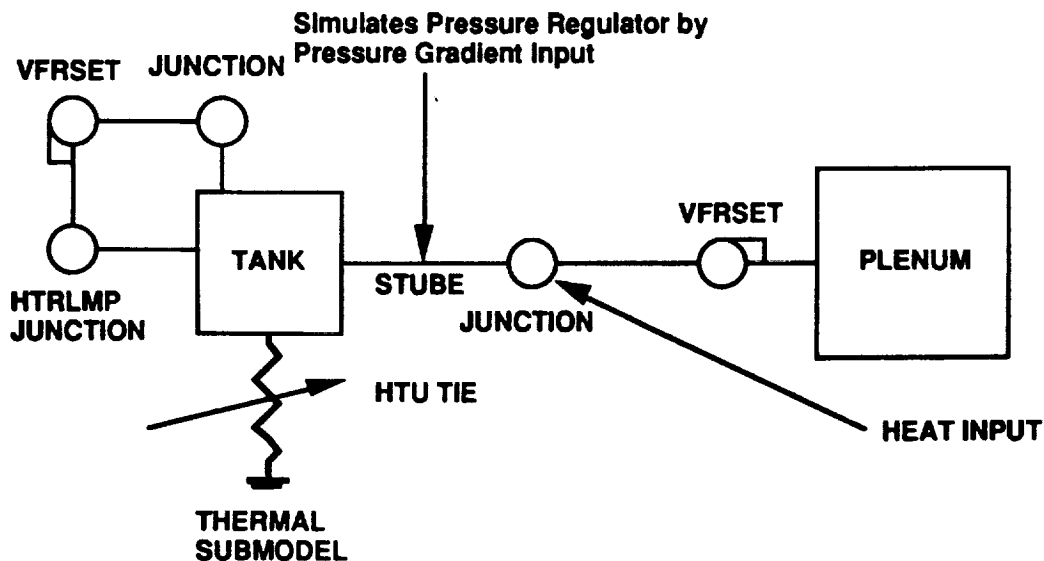


Figure 22: Schematic of the SINDA/FLUINT Airlock Model.

Depressurization Results

The model was first used to predict the thermal response of the airlock gas as it is depressurized from 10.2 psia (70.3 kPa) to 0.5 psia (3.4 kPa). For this situation, three depressurization times were considered: 5.0, 7.5 and 10.0 minutes. Since the hyperbaric heat exchanger does not operate during this phase of airlock operations, the HTRLMP option was taken out of the model. In addition, since convection is miniscule, the value of the heat transfer tie was set to zero.

Figure 23 shows the predicted thermal response of the airlock gas for the three cases with an inlet charge air temperature of 70 °F (294.3 K). Since there is no heat transfer for any of these cases, they all reach the same minimum temperature of -235 °F (125 K). While these low temperatures may be encountered, it should be noted that the crew members will be suited and should remain unaffected.

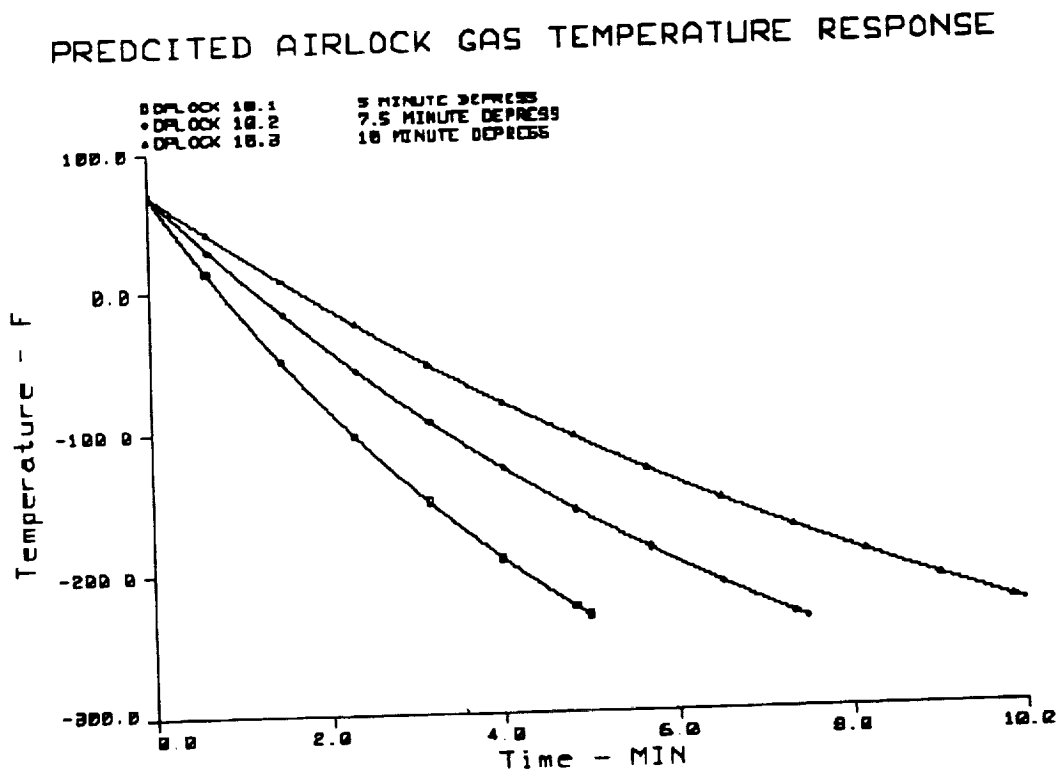


Figure 23: Predicted Airlock Depressurization Gas Temperature Response

Hyperbaric Results

The model was next used to determine the thermal response of the airlock gas during hyperbaric pressurization to 2.8 atmospheres for three different cases with a pressurization rate of 13.2 psi/min (91.0 kPa/min), 70 °F (294.3 K) charge air and an equipment heat load of 1440 Watts. The first case considered an adiabatic situation. The second case included convection with the side walls. Finally, the last case considered the combined effect of the heat exchanger and convective heat transfer.

Figure 24 shows the predicted temperature response of the airlock gas for the three test cases examined. These results show that heat transfer effects can substantially reduce gas temperatures. Specifically, when convection is include the maximum temperature is reduced by approximately 50 °F (28 K) while the operation of the heat exchanger reduces the maximum temperature another 50 °F (28 K) to a comfortable range.

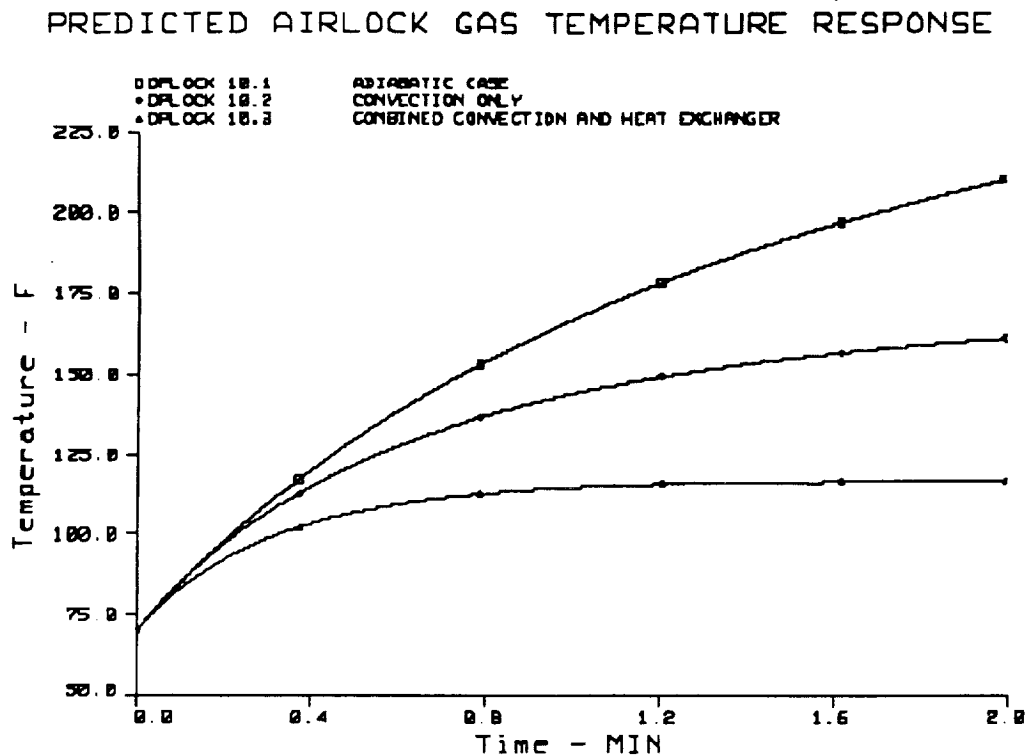


Figure 24: Predicted Temperature Response for Three Different Heat Transfer Cases

During normal operations, the charge air tanks will be heated and cooled by radiative heat transfer with the environment. It is estimated that due to this heat transfer, the temperature of the charge air may vary from 70 °F (294.3 K) to 0 °F (255.4 K). As indicated by equation [2], for a filling process, the final temperature is influenced by the incoming gas temperature, so it follows that charge air temperature will be important during hyperbaric operations. To examine the effect of charge air temperature on system response, four inlet air temperature were used: 10 °F (261 K), 30 °F (272.1 K), 50 °F (283 K), and 70 °F (294.3 K). The charge air was used to pressurize the chamber to 2.8 atm at 13.2 psi/min while the heat exchanger was operating.

Figure 25 presents the thermal response of the airlock gas during pressurization for the four charge air temperatures. From these results it is clear that the temperature of the incoming air plays an important role in chamber temperature response.

Airlock Model Summary and Conclusions

From this study, it was found that convective heat transfer is important in determining airlock gas temperature. By using the heat exchanger, the severity of the high temperatures associated with the compression processes will be lessened. Finally, depress temperatures are low but should be no problem for suited astronauts.

The Simplified Aid for EVA Rescue Device

During the construction phase of the Space Station, astronauts will be required to perform many long-duration EVAs. While conducting these EVAs, it is possible that an astronaut may become separated

from the SSF tethers and float free. Since the Space Shuttle cannot maneuver safely in this construction environment and retrieve the wayward astronaut, an alternative rescue approach must be employed.

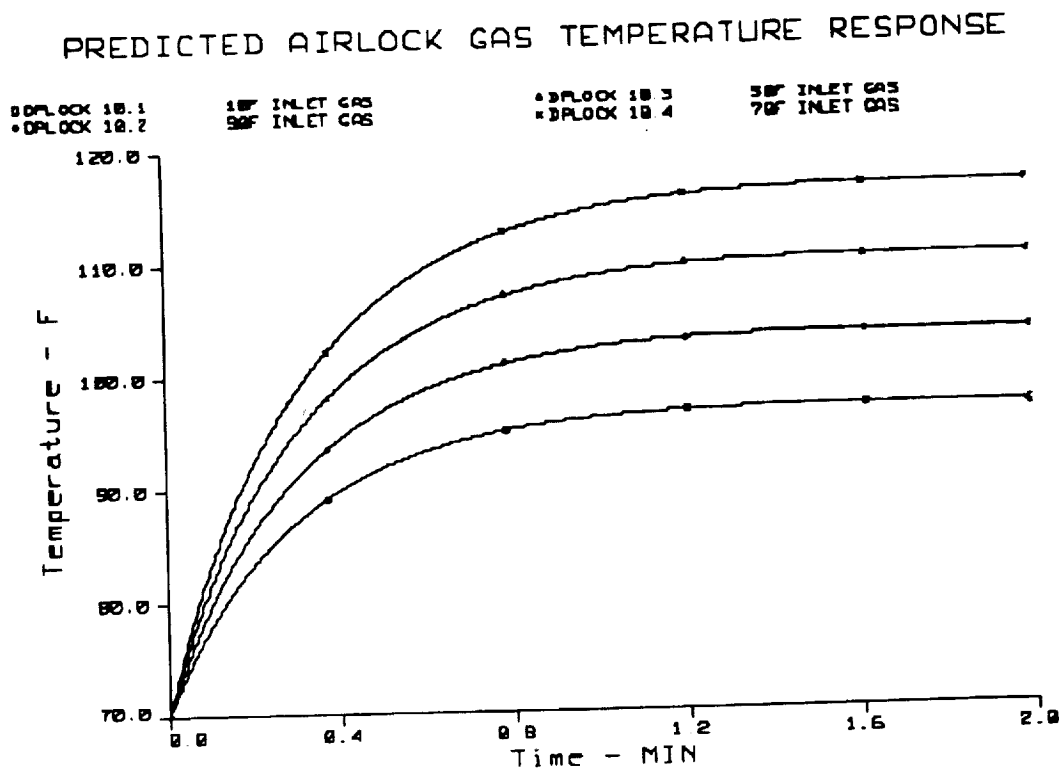


Figure 25: Predicted Response for Four Different Inlet Gas Temperatures

One possible rescue method employs the Simplified Aid For EVA Rescue (SAFER) system. Here, before the EVA begins, a small thruster back pack is attached to the rear of the astronaut's space suit, and then worn throughout the duration of the space walk. This back pack consists of four main propulsion thrusters and 12 smaller course correction nozzles, all of which are fed high pressure, non-reacting gas from a storage tank. Currently, the supply gas is xenon, since it requires the smallest containment vessel compared to other candidate gases. Due to its high molecular weight, gas leakage is minimal (ref. 10). If, during the EVA, an emergency arises, the SAFER system is activated, the astronaut fires the thrusters and guides himself to a place of safety.

As can be shown by the governing equations for supersonic nozzle flow (ref. 11), the performance of the SAFER thrusters depends almost exclusively on the thermodynamic state and properties of the working fluid at the inlet of the propulsion nozzles which is supplied from the storage tank. Since mass is removed from the storage tank during SAFER operation, expansion effects will cause the state within the storage tank to change, ultimately affecting the inlet conditions to the nozzles. In addition, the gas storage bottle radiates to deep space and cools (it may lose heat for six to ten hours before the the thrusters are fired), creating lower tank pressures and temperatures which affect nozzle performance when the thrusters are required. As a result (and as will be shown), the choice of fluid and thermal response of the storage tank have a significant impact on the performance, size and weight of the SAFER system.

Development of the Numerical Models

The SAFER system presents several unique heat transfer and thermodynamic situations which do not lend themselves to simple closed-form solutions. First, the transient cooling of the storage tank by

radiative heat transfer to deep space is described by a non-linear differential equation, and except for a few idealized cases cannot be solved. The SAFER system cannot be treated as an idealized thermal radiator, since it also exhibits conduction and re-radiation, complicating the problem and making an analytical solution impossible. Second, the propulsion thrusters are supersonic nozzles and their performance is determined by nozzle inlet and storage tank conditions, but the storage tank pressure and temperature are dependent upon the mass flow rate out of the thrusters. It is clear then from the above examples, that both situations are quite complex and numerical models must be developed.

Storage Tank Model

The first situation to be examined was the radiative cooling of the storage tank gas before the thrusters are fired, since these results are important in the development and analysis of the thruster model. This model considers the heat loss of the gas and gas containment system to deep space during the EVA, before any propellant is used.

Figure 26 presents a simple schematic of the propellant tank. Here, the xenon gas is contained in a rigid metal pressure vessel, which is protected from meteoroid impacts by a thin metal shield. The entire apparatus is then placed in a holding mount within the plastic (maybe fiberglass) SAFER shell and surrounded with Multi-Layer Insulation (MLI). The gaps between the spheres may be evacuated or filled with MLI. The dimensions listed on this figure are only preliminary (ref. 11) since many factors (choice of nozzle, amount of line heating, amount of gas), which have yet to be accurately determined, influence the tank size.

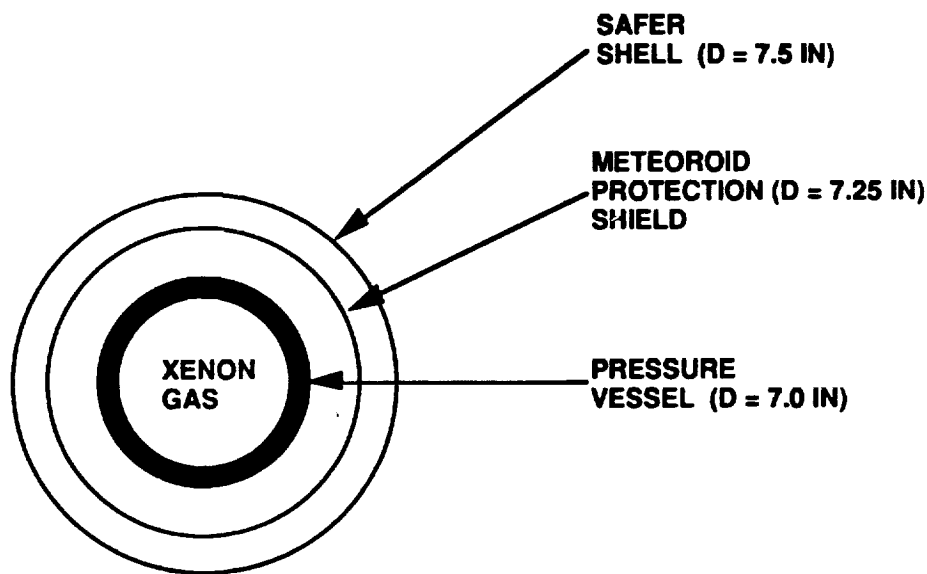


Figure 26: Schematic of the Storage Tank

Before the numerical model was developed, certain simplifying assumptions were made. Assumptions were made for both the gas and its holding vessel and these are listed below.

- 1) The pressure vessel and meteoroid shield are constructed out of stainless steel.
- 2) Xenon is the only gas considered. The model will be developed so that other gases can be considered.
- 3) Preliminary weight estimates are; 16.0 lbm (7.25 kg) for xenon, 10.6 lbm (4.80 kg) for the pressure vessel, and 0.5 lbm (0.23 kg) for the meteoroid shield (ref. 11).
- 4) The pressure vessel, meteoroid shield and SAFER shell are treated as concentric spheres. The conduction and radiation conductors for this situation are found by the method outlined in reference 9.

- 5) Only the worst case cooling situation will be considered. That is, there is no incident radiation from the Sun, SSF, Space Shuttle or Earth on any portion of the SAFER system.
- 6) MLI is used as the insulating material.
- 7) When appropriate, there is no conduction between spheres.
- 8) No fluid is withdrawn from the tank, since this model examines the quiescent fluid before thruster firing.
- 9) Radiating surfaces have an effective emissivity of 0.05.

The development of a SINDA/FLUINT model for the storage tank is a relatively simple task. First, each piece of the storage tank assembly is considered a model node and its thermal capacitance is determined (mass multiplied by specific heat). Next, the heat transfer conductors between the each node are determined by using the methods outlined in References 1 and 7. With this accomplished, a boundary condition (node) is set for deep space at -460°F (-273.2 K) and the FWDBCK method (ref. 1) is used to determine the transient cooling process.

Storage Tank Model Results

In order to understand what factors influence the radiative cooling of the storage tank and to suggest appropriate control methods, several test cases must be examined. For the present study, five cases were examined. Four of the five cases considered radiation as the sole means of heat transport to the deep space environment. That is, the heat transfer between the containment spheres occurs only by thermal radiation. Using Figure 26 as a guide, the four radiation cases considered are: xenon only (Case 1), xenon and the pressure vessel (Case 2), xenon, the pressure vessel and the meteoroid shield (Case 3), xenon, the pressure vessel, the meteoroid shield and the SAFER shell (Case 4). Case 5 considers the same conditions as Case 4 except the evacuated spaces between the spheres is filled with MLI (12 layers) which has an effective emissivity, ϵ^* , of 0.05. For all cases, heat is eventually rejected to the cold environment of space by radiation, over a maximum period of 16 hours.

Figure 27 presents the transient thermal response of the storage tank gas for the five insulating cases as they radiate heat to space over a period of 16 hours. These results are also summarized in Table 5. As can be seen, Case 1 (xenon only) exhibits substantial and unacceptable cooling; however, with the inclusion of the pressure vessel (the minimum design requirement), the temperature drop is severely reduced, since there is additional mass which must be cooled. The use of the meteoroid shield substantially reduces the heat loss, since it acts as a thermal radiation shield (heat transfer barrier). Similarly, the inclusion of the SAFER holding shell further reduces the heat losses. When the entire system is considered, with the MLI included, the heat leak is minimal (4°F (2.2 K) after 16 hours), indicating that a passive scheme can provide acceptable results and eliminate the need for a heater on the pressure vessel.

The Storage Tank/Thruster Model

Figure 28 presents a simple schematic of the thruster system. Here, gas from the storage tank flows into a pressure regulator to ensure that a constant pressure is maintained at the nozzle inlet over the entire time of thruster operation. The working fluid then flows from the regulator outlet to the thruster nozzles where it is expanded isentropically in a supersonic nozzle to provide thrust for the back pack. Since a thruster design (manufacturer) has yet to be chosen, the exact working pressures, temperatures, nozzle areas and flow rates of this system are not known.

Before the thruster model was developed, the following simplifying assumptions were made:

- 1) All the candidate gases (argon, xenon, nitrogen) are considered ideal, even at the storage tank condition of 8000 psia. At the nozzle inlet (after the regulator) the pressures are low enough (approximately 500 - 1000 psia) that the gases behave ideally and the relationships for ideal supersonic flow may be used. The appropriate governing equations will be presented shortly.
- 2) The regulator performs as an ideal throttling device (enthalpy is constant), thus for ideal gases there is no temperature drop.

Table 5: Predicted Gas Storage Temperatures at Various Times for Different Insulating Conditions with the Initial Temperature Set at 70 °F and the Deep Space Environment Set at -460 °F.

Insulating Condition	Temperature After 6 Hours (°F)	Temperature After 10 Hours (°F)
Case 1	0	-25
Case 2	50	40
Case 3	65	60
Case 4	66	64
Case 5	68	67

- Figure 20 was used to model the SAFER system. The storage tank is represented by the TANK option which allows calculation of transient pressure and temperature changes as mass is removed from the tank. To account for the environmental heat loss, the TANK is tied thermally to the previously developed SINDA model of the storage container by the use of a convection heat transfer (HTU) tie. The

value of this tie is determined by the method outlined in reference 12 for determining convection in exhausting spherical pressure vessels. The regulator is represented using the STUBE option and a junction (JUNC). Since the storage tank pressure changes, user logic is employed in the main program to alter the pressure across the STUBE (the HC term) so that the required pressure is maintained at the junction. This junction also represents the inlet to the nozzle and it is here that inlet heating is applied. The nozzle is modeled using an MFRSET where the mass flow of this device is governed by the following equation,

$$m = A_t * P_o * \sqrt{\frac{k * G_c}{T_o * R} * \left(\frac{2}{k+1}\right)^{\left(\frac{k+1}{k-1}\right)}} \quad [3]$$

where, A_t is the nozzle throat area, P_o is the nozzle inlet pressure, k is the ratio of specific heats, G_c is the conversion factor, R is the specific gas constant, and T_o is the nozzle inlet temperature. Since T_o varies as the storage tank is depressurized the mass flow rate is updated at each iteration. Because A_t and P_o are specific to a given thruster, these quantities are input to equation [3].

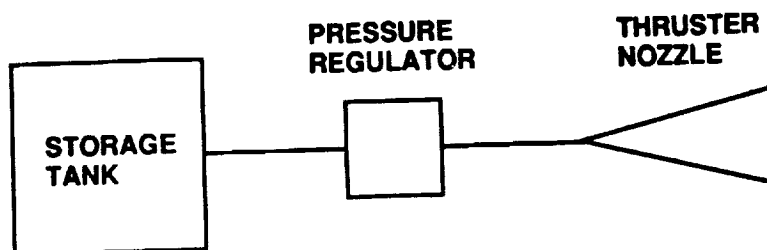


Figure 28: Schematic of the SAFER System.

Results for the Storage Tank/ Thruster Model

For the thruster study, two flow situations were examined. First, the model was run with and without heating of the incoming gas to examine system response. Finally, three candidate gases (argon, xenon, nitrogen) were examined for identical operating conditions to determine their individual effects on the thermal response of the storage tank and system performance. For all cases considered, the initial condition of the storage tank was 8000 psia (55.2 MPa) and 70 °F (294.3 K).

The model was next run to examine the effect of heating the inlet gas on system performance. Figure 29 shows the predicted thruster mass flow rate, with xenon as the working fluid, with and without heating during a one minute operation period. For the heated condition, the nozzle inlet temperature is maintained at the initial conditions of 70 °F (294.3 K), while for the unheated situation, the inlet nozzle temperature is identical to the storage tank temperature. From these results, it is clear that inlet heating is beneficial, since at the end of the run an approximately 30% greater mass flow rate is required to maintain the same performance. In other words, without inlet heating, propellant will be consumed at a faster rate. Of course, this problem could be alleviated by using a larger storage tank; however, the SAFER system would be larger and bulkier.

To further examine what factors influence system performance, the model was next run for three different working fluids for identical operating conditions, including heating of the inlet gas to 70 °F (294.3 K). Figure 30 presents the thermal response of the storage tank for one minute of thruster operation. Here, nitrogen shows a substantially higher tank temperature than the other two gases. Since the thermal response of the gas within the tank is governed by the equation [1], it follows that gases with higher k values will produce lower tank temperatures during depressurization. Since argon and xenon have higher values of k than nitrogen, it follows that their storage tank temperatures must drop more quickly than nitrogen.

PREDICTED FLOW RATES

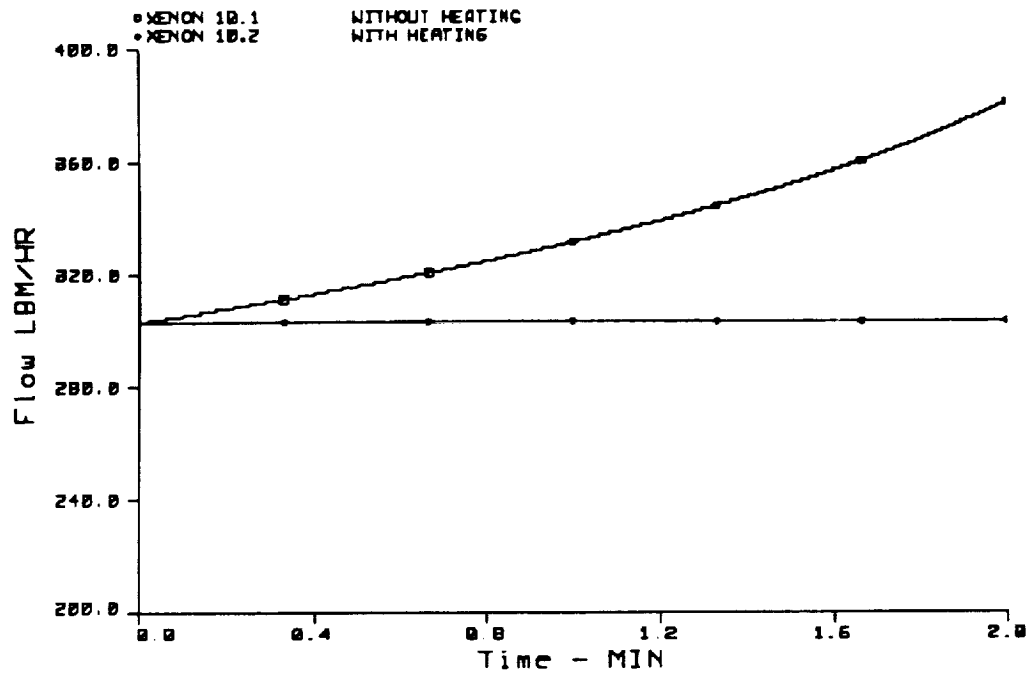


Figure 29: Predicted Mass Flowrates for Heated and Unheated Inlet Gas

PREDICTED STORAGE TANK GAS TEMPERATURE

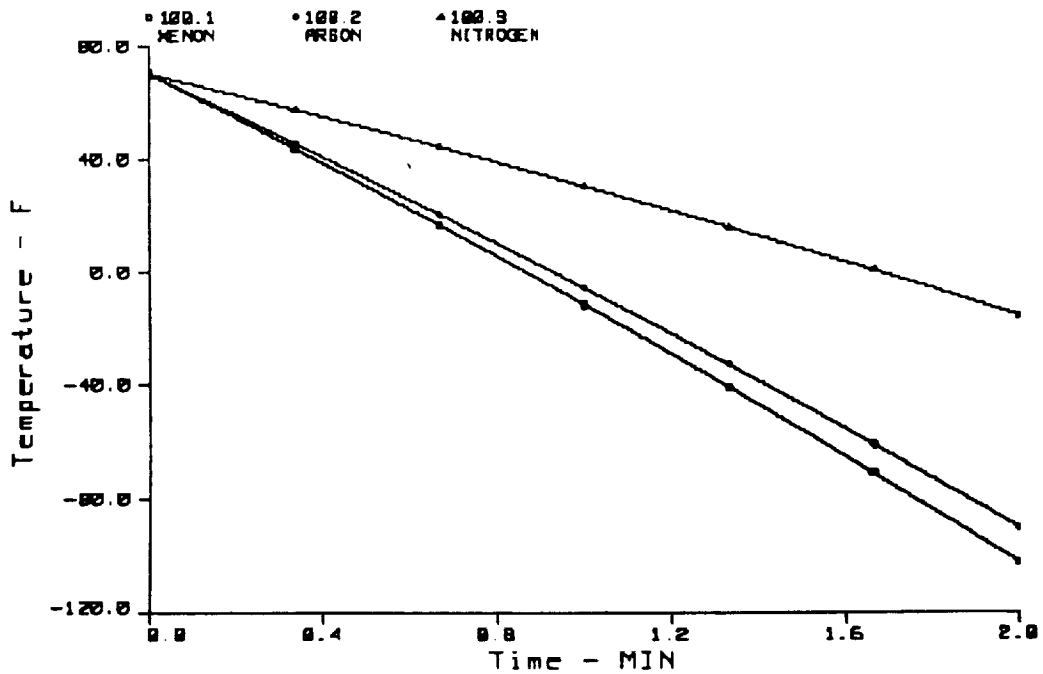


Figure 30: Predicted Thermal Response of the Storage Tank for Three Candidate Gases

While the results of Figure 30 suggest that nitrogen would need the least amount of inlet heating (smallest temperature difference between the initial, 70 °F (294.3 K), and the tank temperature), attention is directed to Figure 31 which shows that over the period of operation, xenon requires the least amount of heating. For this situation, the heating of the gas can be determined from the first law of thermodynamics ($Q = mC_p(70 - T_{\text{Tank}})$) and of the three candidate gases, xenon has the lowest specific heat. As a result, from a heating point of view, it is recommended that xenon be used as the working fluid, since it requires the least amount of heating.

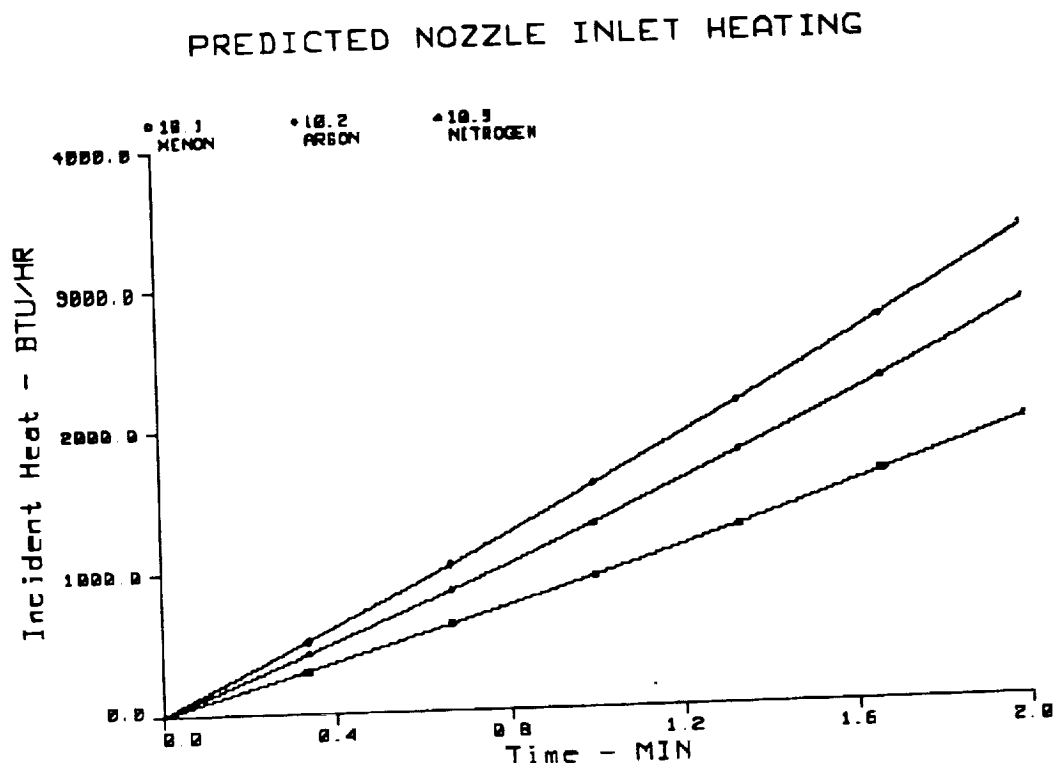


Figure 31: Predicted Heat Addition to Maintain the Inlet at 70°F for Three Gases

SAFER Summary and Conclusions

Two SINDA/FLUINT models have been developed to predict the transient thermal and hydrodynamic response of various components of the SAFER system and serve as a design tool. Since the exact thruster design has yet to be chosen, the models have been built so that they can be easily modified to handle any changes in working fluid, thruster design, insulating materials and pressure regulators.

One model predicts the thermal response of the storage tank as it radiates to space. This model predicts that with a simple MLI insulating scheme and a meteoroid shield, there will be minimal heat leak to the environment during the EVA time (sixteen hours or less).

The other model predicts the thermodynamic performance of the thrusters and the thermal response of the storage tank during thruster operation. The results indicate that heating of the inlet gas is beneficial, by reducing thruster mass flow rates, in turn reducing the size of the tank. In addition, preliminary results indicate that xenon should be chosen as the propellant, since it requires the lowest heat input to maintain the required conditions.

CONCLUSION

Because of its versatility, SINDA/FLUINT has been applied to a wide variety of aerospace problems at JSC and elsewhere. For different problems, different features of the code have been used. Some aspects of the code which have been utilized at JSC include transient simulation, boiling and condensation, two-phase flow pressure drop, slip flow, multiple submodels and depressurization. Several of these applications have been described in this paper. Use of SINDA/FLUINT by the aerospace community and others continues to increase as the capabilities of the code expand (ref. 13).

REFERENCES

1. Cullimore, B., et al., SINDA/FLUINT Systems Improved Numerical Differencing Analyzer and Fluid Integrator, Version 2.4, User's Manual, December 1991.
2. Rohsenow and Harnett, Handbook of Heat Transfer, 1973.
3. Shah, M. M., 'A General Correlation for Heat Transfer During Film Condensation Inside Pipes', Int. Journal of Heat & Mass Transfer, Vol. 22, pp 547-556, 1979.
4. DeMarchi, D., 'Freeze Prevention of Ammonia in the CTB Radiators', Lockheed Engineering and Sciences Co. Technical Memorandum, LESC-30053, Jan. 1992.
5. VanWylen, G.J., and Sonntag, R.E., Fundamentals of Classical Thermodynamics, 3rd ed., John Wiley and Sons, New York, 1986.
6. Loomis, A.W., Compressed Air and Gas Data, Ingersoll-Rand, New Jersey, 1980.
7. Incropera, F.P., and DeWitt, D.P., Fundamentals of Heat and Mass Transfer, John Wiley and Sons, New York, 1985.
8. Kays, W.M., and London, A.L., Compact Heat Exchangers, 3rd ed., McGraw-Hill, New York, 1984.
9. "Analysis Report - Hyperbaric Chamber Heat Loads and Removal Methods," prepared by Eagle Technical Services Inc., Houston, Tx. for Lockheed Missiles and Space Company, Inc., Houston, Tx.
10. SAFER Pre-Preliminary Design Review, May 1991.
11. Shapiro, A.H., The Dynamics and Thermodynamics of Compressible Fluid Flow. Vol. 1, John Wiley and Sons, New York, 1953.
12. McKinney, P.H., "An Investigation of the Thermodynamic Performance of a Discharging Spherical Pressure Vessel", Ph.D. Thesis, Rice University, 1973.
13. Cullimore, B. A. , Ring, S. G. and Ungar, E. K., "Development Status of SINDA/FLUINT and SINAPS", Fourth Annual Thermal and Fluids Analysis Workshop, NASA Lewis Research Center, August 17-21, 1992.

TWO-PHASE/TWO-PHASE HEAT EXCHANGER SIMULATION ANALYSIS

Rhyn H. Kim
Department of Mechanical Engineering
The University of North Carolina at Charlotte
Charlotte, N.C. 28223

SUMMARY

The capillary pumped loop (CPL) system is one of the most desirable devices to dissipate heat energy in the radiation environment of the Space Station providing a relatively easy control of the temperature. A condenser, a component of the CPL system, is linked with a buffer evaporator in the form of an annulus section of a double tube heat exchanger arrangement: the concentric core of the double tube is the condenser; the annulus section is used as a buffer between the conditioned space and the radiation surrounding but works as an evaporator. A CPL system with this type of condenser is modeled to simulate its function numerically. Preliminary results for temperature variations of the system are shown and more investigations are suggested for further improvement.

INTRODUCTION

A capillary pumped loop (CPL) system consists of an evaporator pump, a condenser, a subcooler, a vapor transport line, a liquid transport line and a storage tank with a possible starter pump (Ku and others, 1986; Neiswanger and others, 1987; Kim, 1990). The condenser structured as a concentric core of a double tube heat exchanger dissipates heat energy to an evaporator, which is the annulus section of the double tube exchanger, so that the fluid in the core condenses while the fluid in the annulus section evaporates. A CPL system equipped with a condenser having a two-phase/two-phase double tube heat dissipation structure is to be simulated numerically. A schematic block diagram is shown in Fig.1 and illustrates the control devices of its flow.

The system has two different types of evaporator pumps, each of which has twelve units of evaporator pumps, double two-phase heat exchanger (call DBTPHX) system, which has six DBTPHX units as shown in Fig.2, a subcooler next to the DBTPHX system with a non-condensable gas collector, long liquid and vapor lines, a sub-system of control devices consisting of valves and fluid meters and a reservoir with a starter pump. In order to have a manageable simulation system, certain assumptions are made: (1) A cold plate evaporator pump replaces a hybrid evaporator so that two identical cold plate evaporator pumps are a set of evaporator pumps; (2) fan-shape inlet and exit sections of the DBTPHX (Fig.2) are assumed to be an assembly of six straight tubes in the same level of elevation; (3) the non-condensable gas(NCG) collector in the subcooler does not affect the performance of the system; (4) the flow control devices for the system is not included in the model in question; and (5) the storage tank behaves as an infinite source with constant properties. Such assumptions result in the block diagram as shown in

Fig.3.

Starting from the end of the subcooler, a few key points are identified as J1001, J1002,...,J1005 along with the Line 101, Plen 999, evaporator, Line 401, TEMP2C condenser, TEMP3C evaporator, TEMP2C subcooler and R114 chiller. A code known as SINDA'85/FLUINT (Cullimore and others, 1989) is employed for the simulation scheme and similar notations to those of SINDA'85/FLUINT are used in Fig.3. Details of these components consisting of an entire simulation model are illustrated in the following sections.

EVAPORATOR PUMP SYSTEM

Cross-sections of the evaporator pump in two directions are shown in Figs.4a and 4b. Fig.4c illustrates how the liquid flow evaporates. The cold plate evaporator pump(EVP) with twelve(12) EVP's, twelve isolators are shown in Fig.4d, and a simplified model is indicated within the cross-hatched area in Fig.4d with one unit of the heat-pump and an isolator attached to it. Fig.4e explains the cross-section of the circular heat-pump and attached plates. Based on this figure of the combined cross-section, an approximate method for the extended surface theory is used. Namely, at the mid point of the plate, the temperature is the maximum and the temperature at the tube is a fixed temperature. Therefore, an approximate fin efficiency is used for an average temperature(or conductance) as far as the plate is concerned. The radial cross-section of the EVP has 40 internal grooves in the outer shell. From a header, liquid enters the isolator, permeates the porous layer, reaching the core of the EVP. Once the liquid reaches the groove surface through permeation, heat transfer from the outside causes vaporization of the liquid. Evaporated fluid is pushed to the grooves(Fig.4c) and to the vapor header and enters the vapor transport line. Cullimore (1989) successfully demonstrated a numerical model by using a MACRO command, CAPPMP. Therefore, his method is employed for the EVP system. The capacity of the EVP system is assumed to be 400 watts.

DOUBLE TWO-PHASE HEAT-EXCHANGER SYSTEM

In the DBTPHX, the condenser is a inner circular tube coupled with an evaporator which is the annulus section of the DBTPHX. The inner tube has axial grooves internally and externally and porous material layers occupy the space next to the grooves in the annular section and the core section, respectively, so that liquid, from capillary action can permeate the grooves and the porous layers.

Six DBTPHXs connected in parallel function as the condenser and are designated as TEMP2C. The layer of porous material, Porex, enhances condensation in the internal grooves. Six DBTPHX evaporators connected in parallel are designated as TEMP3C. The layer of Porex directs the liquid flow in one desired direction. Porex has a permeability of 2.3×10^{-13} .

From the inlet of TEMP2C a vapor enters internal grooves of the internal tube, condenses on the surface of the porous layer and permeates the porous layer, reaches the core cavity and leaves the TEMP2C from the core section of the DBTPHX. From the inlet of TEMP3C (the exit side of TEMP2C), liquid flows into the annulus section where the porous layer is placed. The porous layer has grooves in the metal outer shell side. As the liquid permeates the porous layer, it reaches the external grooves of the internal tube, starts evaporating because of heat transferred from the condenser, then is pushed into the grooves, leaving TEMP3C from the annulus section. From this exit, it is possible for a mixture of saturated liquid and vapor to leave TEMP3C.

Figs. 5a and 5b show an axial cross-section and a radial cross-section, respectively. Six of the fan-shaped inlet and exit sections are represented by a single straight tube of the same size at the same elevation then duplicates six times. The core section from which the liquid leaves TEMP2C has an isolator then a liquid header follows. Due to heat transfer from TEMP2C to TEMP3C across the metal tube, it is plausible to consider that the quality of condenser fluid changes from unity to zero, while the quality of evaporator fluid changes from zero to unity. For simplicity, the variations of fluid quality in TEMP2C and TEMP3C are assumed to be linear as shown in Fig.6. To the mid point of TEMP2C, and TEMP3C, the qualities remain unchanged then change linearly to the locations from the mid point of the DBTPHX. In the present study, the length of the DBTPHX is divided into ten segments. Accordingly, the quality will be assigned for TEMP2C and TEMP3C.

SUBCOOLER AND R114 CHILLER

The subcooler of the CPL has a complicated structure for a vapor trap to cope with non-condensable gas in the subcooler. Considering that the amount of noncondensable gas is relatively small in comparison with the flow rate, the vapor trap is not modeled in the present study. In the location of the trap, a flat plate is placed. Similarly to the evaporator pump plate, the half of the plate with the attached subcooler and the R114 chiller is considered to be like a fin having the minimum temperature at the mid point of the contacting area between the subcooler and the chiller (see Fig.7). The fin efficiency is assumed to be 80 per cent. The temperature variation along the transverse direction is assumed to be rather small in comparison to that in the direction of the tube axis. For each leg of the subcooler and the chiller a uniform temperature is assumed to exist and the 180 degree bends connecting four legs are considered to be adiabatic. This is to be handled with a MACRO command of SINDA'85/FLUINT.

The subcooled liquid passes through the liquid transportation line which is considered adiabatic. A pressure drop through this passage is added to the system pressure loss. The liquid then enters the evaporator pump.

The liquid reservoir is added to the system model as a plenum, holding all the properties as constant. A mixing process in the reservoir may not result in constant properties, nonetheless, it is assumed a steady state process.

INPUT FILE FOR THE COMPUTER PROGRAM

An input file for SINDA'85/FLUINT was constructed based on the following Headers:

1. Header Options Data
2. Header Node Data,Plate
3. Header Conductor Data,Plate
4. Header Control Data
5. Header Source Data,Plate
6. Header Flow Data,TEMP2C
7. Header Flow Data,TEMP3C
8. Header Flow Data,STLTS
9. Header Flogic 0,TEMP2C
10. Header Flogic 0,TEMP3C
11. Header Flogic 1,TEMP2C
12. Header Flogic 2,TEMP2C
13. Header Variables 1,Plate
14. Header Output Calls, TEMP2C
15. Header Operation Data
 - Build DBTPHX,Plate
 - Build DBTPHX,TEMP2C,TEMP3C,STLTS
 - Call Fastic
 - Call Fwdbck
16. Conditional Call for Restar
17. Header Subroutine Data if any
18. End of data

Details of these headers are explained in the manual(Cullimoreand others,1989). Diffusion submodels consist of the evaporator tube, its plate with the web,the internal tube, the external tube of the DBTPHX and its subcooler and chiller bodies with the plates. All of these submodels are represented by a diffusion model, PLATE. Node and Conductor Data sections have the initial temperature,capacitance and conductance of these submodels. The fluid submodels are TEMP2C, TEMP3C,and STLTL. TEMP2C represents the evaporator pump, condenser, its subcooler and the transportation lines. TEMP3C represents the buffer evaporator and STLTL represents the refrigerant chiller. Transportation lines for liquid and vapor are adiabatic. The numbers for nodes,conductors, lumps and connections of the EVP's are in 200's and the remaining numbers used for the thermal and fluid submodels are listed in Table 1.

Using the standard notation for SINDA'85/FLUINT, one unit of EVP is illustrated in Fig. 8 by using the notations of the code, whereby the macro command CAPPMP is applicable.

A symbolic diagram is drawn for the condenser/TEMP2C, the evaporator/TEMP3C, the subcooler/TEMP2C, and the R114 chiller in Fig. 9 with an aid of the standard notations of SINDA'85/FLUINT for DBTPHX. The fluid transportation lines are modeled by a MACRO command LINE. The subcooler/R114 chiller is modeled by a MACRO command HX.

Table 1 Diffusion and Fluid Model Identifications
by a Range of Numbers

Liquid Transportation Line	100 to 199
Evaporator	200 to 299
Vapor Transportation Line	400 to 499
Condenser/TEMP2C	500 to 599
Subcooler/TEMP2C	600 to 699
Evaporator/TEMP3C	700 to 799
R114 Chiller	800 to 899
Storage Tank	900 to 999

RESULTS and DISCUSSIONS

A start-up performance is run as the first trial with a prepared file shown above. With a surrounding temperature of 6.6 C, every component of the CPL system is at the same temperature except the storage tank which is at 29 C. A heat load of 400 w is applied to the cold plate EVP, and for a certain time period, performance of the system is simulated. Computation is divided into two time zones: the first 2 minutes and the following 28 minutes. In the 2 minute duration, the first step of the calculation was Fastic's procedure to obtain a stable initial condition for the entire system. Thus, reaching a stable condition, the Fwdbck procedure takes over the computation. The Fastic scheme provides computations in an instantaneous equilibrium. The Fwdbck process involves an implicit temperature expression in the way of the Crank-Nicolson(1947) computation process(Cullimore and others,1989). This way, temperature-histories at the EVP plate, the DBTPHX plate and the chiller are plotted for the first 2 minutes in Fig.10. Fig.11 exhibits temperature histories at those locations for the following 23 minutes with the FWDBCK scheme. Some of the results by Neiswanger and others(1987) seem to show that the trend of temperature rise is similar.

At about 25 minutes, a steady state is reached: the highest temperature at the evaporator plate is 39.7 C and the fluid temperature of the evaporator pump is 38.4 C. This tendency should be compared with that of an experimental result if any.

Results of shorter segments of DBTPHX were not included and the accuracy of the current result has not been established with respect to the segment size. Other parameters like the pressure, quality, heat transfer also have not been included. Other operating conditions such as starting from a usual standard condition and a combination of Fwdbck initially and STDSTL operations have not yet been tested with this input file.

These analyses are ongoing at present and will continue until satisfactory results are obtained within a reasonable error limit.

In the process of computation, it was necessary to use the Fastic scheme initially with the linear relationship for the quality along the half length from the mid point in both TEMP2C and TEMP3C sides of DBTPHX. As a result, a stable result was obtained within a few number of iterations in the Fastic calculation. Thereby, the Fwdbck procedure was carried out further, eventually to lead to a steady state condition.

CONCLUDING REMARKS

A CPL system with a DBTPHX condenser was analyzed to simulate performance of the system with simplifying assumptions. The maximum temperature at the evaporator plate reached 39.7 C. Further parameter studies should be done for better results of temperature, pressure and quality distributions. No experimental results have been compared to the present results. That comparison will be critical for refining the input file.

In general, improvements can be made by increasing the number of the segments for the DBTPHX for more accurate solution, because the process of Fwdbck gives stable solutions but does not provide accurate solutions. Similarly, the evaporator pump and the chiller legs should be divided into several segments and search for better solutions for performance simulation is desirable. Other modeling methods for the DBTPHX may be tested and their results should be compared with the result presented here.

The effect of the neighboring isolators on the liquid flow, and precise heat convection coefficients at the evaporating and condensing surfaces in the grooves and porous layers are not available. Those values that come from the subroutine are the convection coefficients for the usual boiling and condensing conditions over a flat plate or cylinder (ASHRE, 1989; Chen, 1963). The heat transfer coefficients used in this paper may yield first order approximate performances, however, the average heat transfer coefficient obtained from the NASA experiment (Neiswanger and McIntosh, 1987; Cullimore, 1989) will be used for further testing. Finally, deleting the simplifying assumptions in making a working model will yield a satisfactory simulation.

ACKNOWLEDGEMENT

This project was supported by the GSFC, NASA (Grant Number #NAG5-1503) and special appreciation is extended to the Program Officer, Mr. T. Swanson for providing valuable reference materials developed by OAO corporation, Greenbelt, Md.

REFERENCES

1. ASHRAE Hand Books, Fundamentals, 1989, I-P Edition.
2. Chen, J.C., 1963, "A Correlation for Boiling Heat Transfer to Saturated Fluids on Convective Flow, "ASME Paper 63-HT-34
3. Crank, J., Nicolson, P., 1947, "A Practical Method for Numerical Evaluation of Solutions of P>D>E of Heat Conduction Type, "Proc. Camb. Phil. Soc., Vol.43, p.50
4. Cullimore, B. et al., 1990, Systems Improved Numerical Differencing Analyzer and Fluid Integrator, User's Manual, Version 2.3, Contract NAS9-17448, MCR-90-512, Martin Marietta, Denver, Colorado
5. Cullimore, B., 1991, SINDA/FLUINT, The Second NASA Thermal/Fluid Workshop Note, Sponsored by the Lewis Research Center, NASA, Cleveland, Ohio
6. Kim, R., 1990, Two-Phase/Two-Phase Heat Exchanger Analysis, Inter. Report to NASA/ASEE Summer Faculty Research Program, Sponsored by the NASA, GSFC, Greenbelt, Md. and Howard University
7. Ku, J., Kroliczek, E., Butler, D., McIntosh, R., and Schweickart, 1986, "Capillary Pumped Loop Gas and Hitchhiker Flight Experiments," Paper No., 86-1248, AIAA/ASME 4th Thermophysics and Heat Transfer Conference, Boston, Massachusetts, June 2-4
8. Neiswanger, L., McIntosh, R., 1987, CPL Modeler, Version 1.2, NASA, GSFC, Greenbelt, Md., June 1987

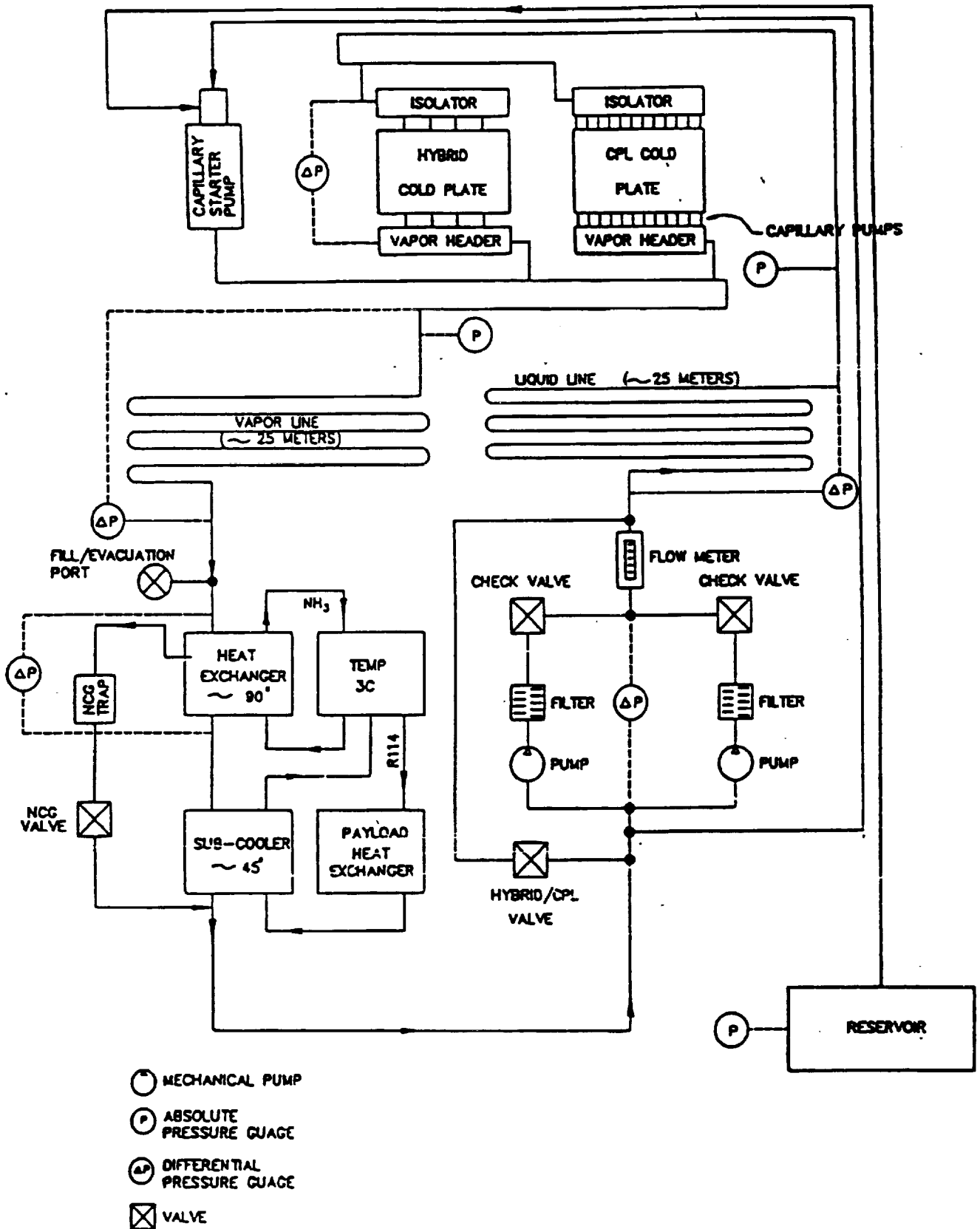


Fig.1 A Capillary Pumped Loop System with a Double Two-Phase Heat Exchanger

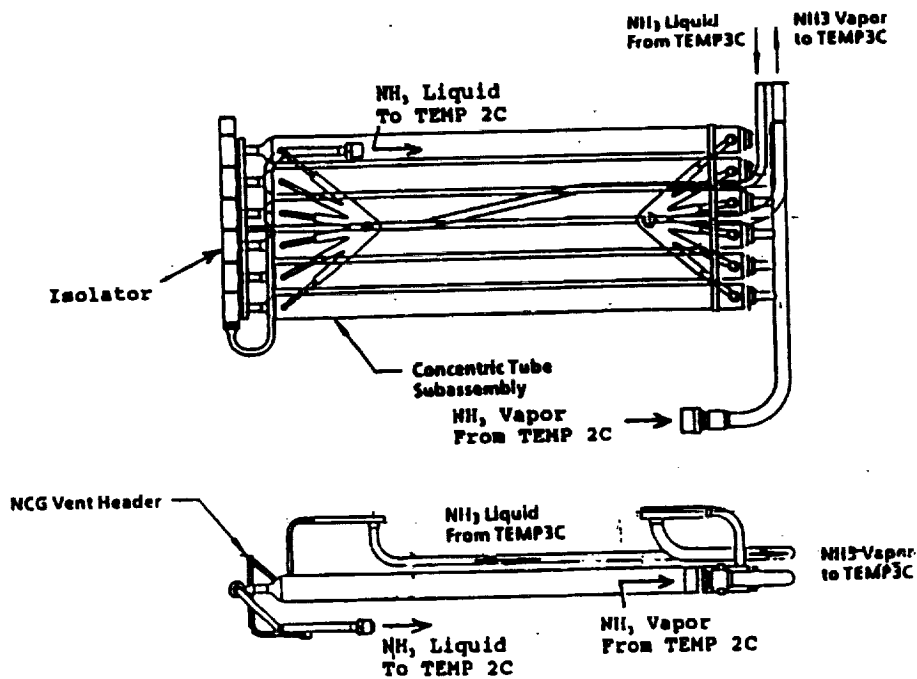


Fig. 2 A Double Two-Phase Heat Exchanger and its Front and Elevation Views

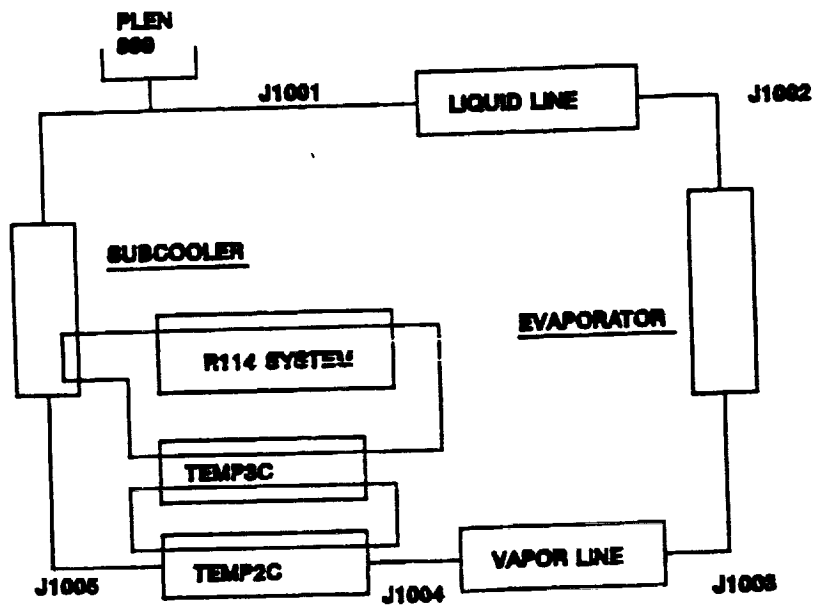


Fig. 3 A Simplified Block Diagram for the CPL System

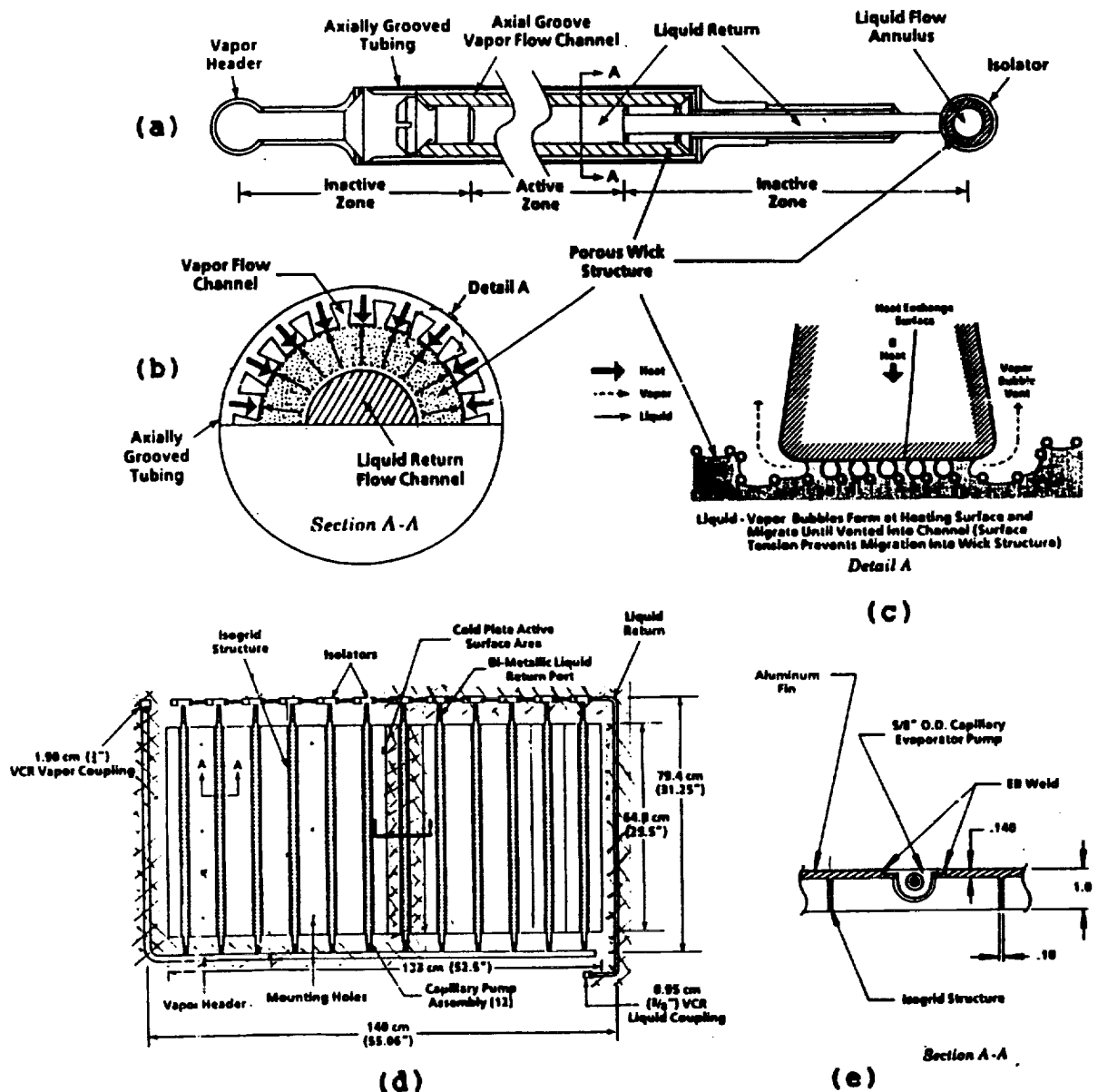


Fig.4 A Cold Plate Heat Pump Evaporator System(a)Axial Cross-Section(b)Radial Cross-Section of Heat Pump Unit(c)Vapor-riazation at the Groove Surface(d)Twelve Heat Pumps,its Isolators and a Simplified Model in the Shaded Area and (e)a Cross-Section of the Circular Heat Pump and Plate

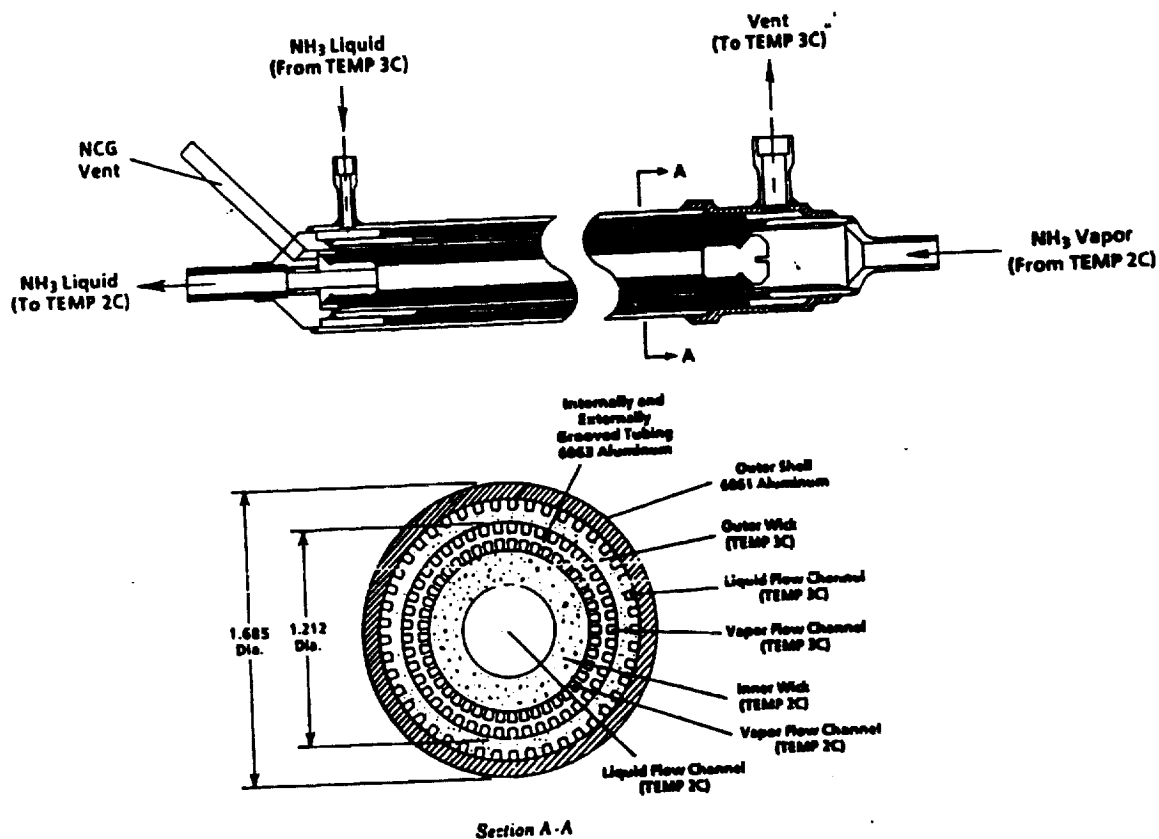


Fig. 5 Double Two-Phase Heat Exchanger/TEMP2C, TEMP3C
 (a) Radial Cross-Section of a DBTPHX
 (b) Axial Cross-Section of a DBTPHX

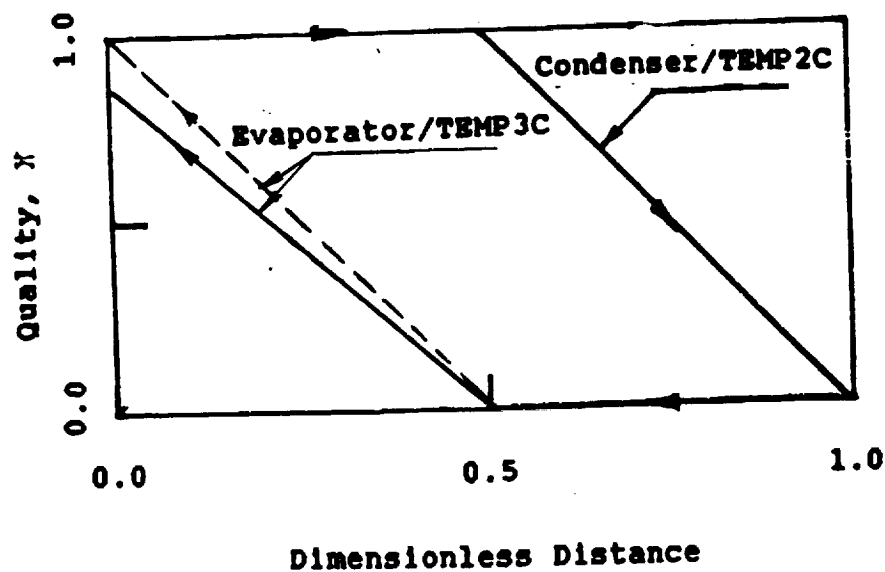


Fig. 6 Initial Quality Variations in TEMP2C and TEMP3C of DBTPHX

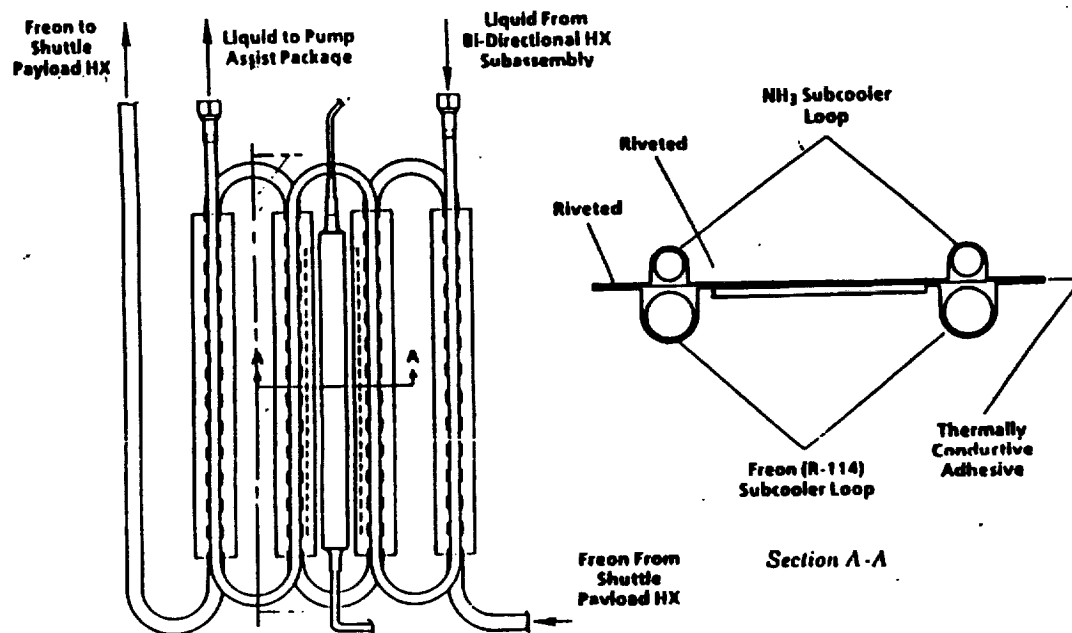


Fig.7 Subcooler/TEMP2C,R114 Chiller and Modified Mid-Section Deleting NCG Trap

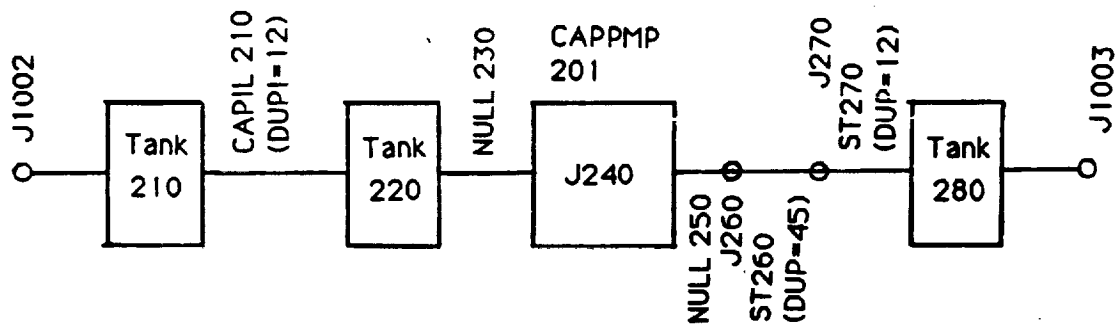
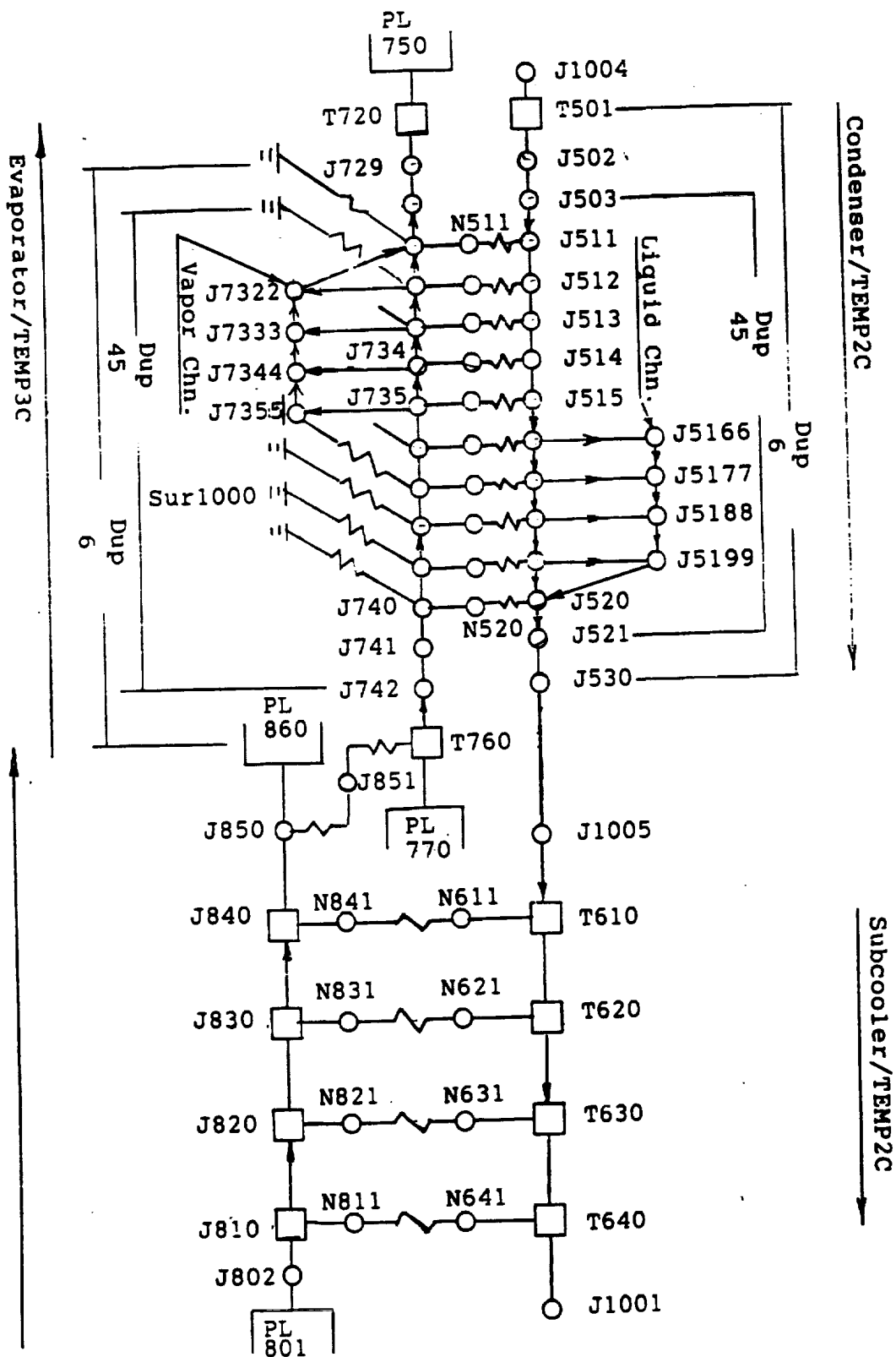


Fig.8 Block Diagram of CAPPMP for the Evaporator System

Fig. 9 Block Diagram of Condenser, Subcooler/TEMP2C, Evaporator/TEMP3C and R114 Chiller



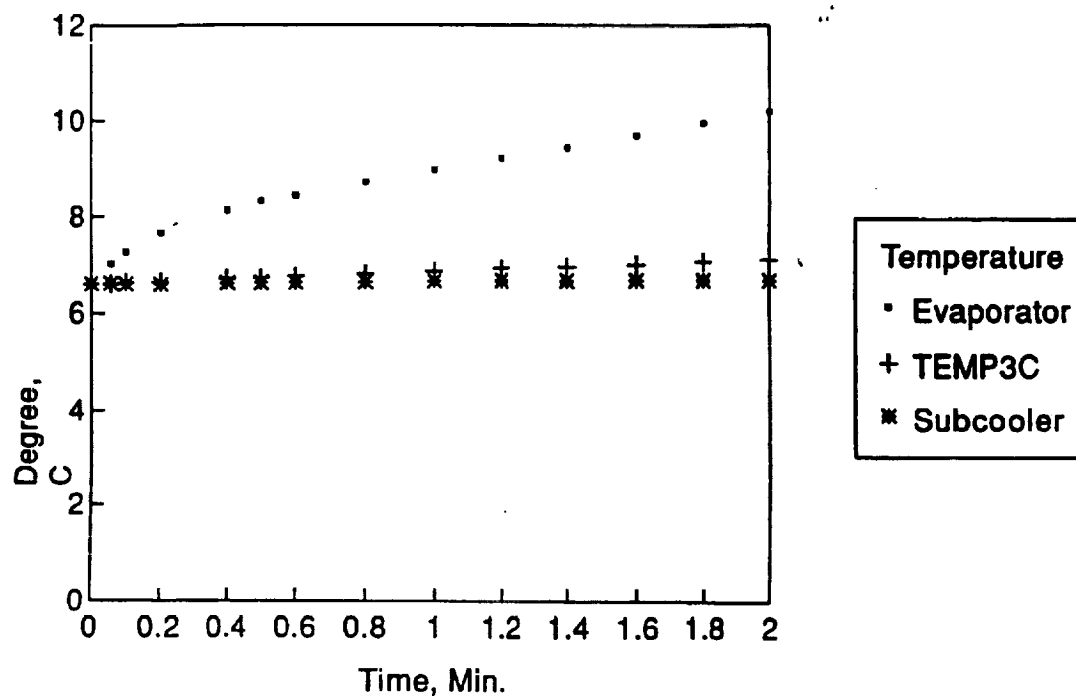


Fig.10 Temperature-Histories at Evaporator,TEMP3C and Subcooler for the First Two Minutes

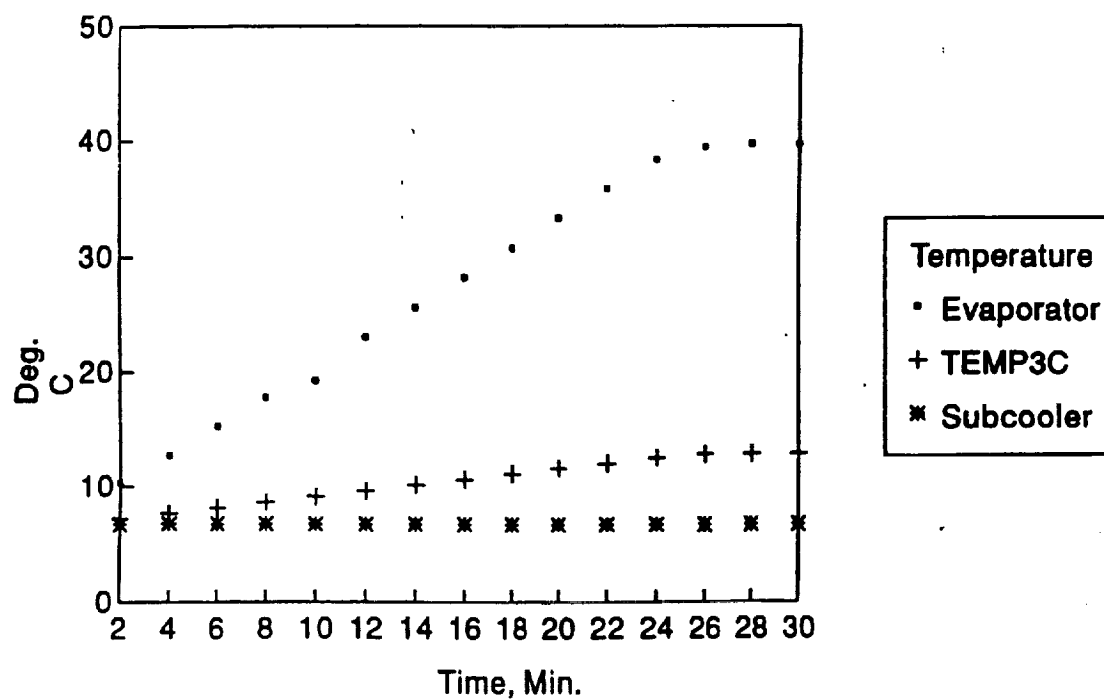


Fig.11 Temperature-Histories at Evaporator,TEMP3C and Sucooler for 30 Minutes

A SINDA '85 Nodal Heat Transfer Rate Calculation User Subroutine

Derrick J. Cheston
 NASA Lewis Research Center
 Cleveland, Ohio 44135

SYMBOLS

C_i thermal capacitance of node i
 G_{ij} linear conductance between nodes i and j
 \overline{G}_{ij} nonlinear conductance between nodes i and j
 Q_i impressed heat on node i
 $Q_{i,linear}$ linear energy transfer rate into node i
 $Q_{i,nonlinear}$ nonlinear energy transfer rate into node i
 $Q_{i,sum}$ total of all energy into node i
 T_i temperature of node i
 t time

SUMMARY

This paper describes a subroutine, GETQ, which was developed to compute the heat transfer rates through all conductors attached to a node within a SINDA '85 thermal submodel. The subroutine was written for version 2.3 of SINDA '85. Upon calling GETQ, the user supplies the submodel name and node number which the heat transfer rate computation is desired. The returned heat transfer rate values are broken down into linear, nonlinear, source and combined heat loads.

INTRODUCTION

SINDA '85 has many powerful subroutines and utilities which easily allow the user to access temperature, capacitance, conductance, and heat source values associated with thermal models. The ability to access these variables to affect the solution is an invaluable attribute of SINDA '85.

Often, however, a user needs the value of the heat transfer rate to a node through all conductors attached. The SINDA '85 Subroutine Library includes subroutines which perform "energy maps" for selected nodes. The usefulness is

limited by the fact the computed heat transfer rate information is output to a file, and thus is not accessible during execution. If a user wants to access the heat transfer rate values they must be calculated by logic input by the user. The user must calculate the temperature difference between nodes and multiply that value by the conductance between those nodes. An increasing

number of conductors attached required a disproportionate increase in logic required.

The user subroutine GETQ, described herein, automated the procedure described above. GETQ computes and returns the value of the sum of energy transfer rates into a node.

GETQ FUNCTIONAL DESCRIPTION

The subroutine GETQ was written to compute and return the value of the sum of the energy transfer rates to a given node. The finite difference form of the energy equation used by SINDA '85 is shown below:

$$C_i \frac{dT_i}{dt} = Q_i + \sum_{j=1}^N (G_{ij}[T_j - T_i] + \overline{G}_{ij}[T_j^4 - T_i^4]) \quad (1)$$

The right hand side of Equation 1 consists of three terms which are described as source, linear, and nonlinear terms. The source term, Q_i represents the impressed heat load onto the node. The linear term, Q_{i_linear} represents the sum of the linear heat transfer rate into the node through all linear conductors. The term $Q_{i_nonlinear}$ represents the nonlinear heat transfer rate into the node through all radiation conductors. The expressions for the linear and nonlinear terms are shown below in equations (2) and (3) respectively:

$$Q_{i_linear} = \sum_{j=1}^N (G_{ij}[T_j - T_i]) \quad (2)$$

$$Q_{i_nonlinear} = \sum_{j=1}^N (\overline{G}_{ij}[T_j^4 - T_i^4]) \quad (3)$$

The GETQ subroutine returns the values of Q_i , Q_{i_linear} , $Q_{i_nonlinear}$ and the sum of these three values, Q_{i_sum} .

GETQ USAGE INSTRUCTIONS

The version of GETQ presented herein is written to be used as a user subroutine. It could, with proper modification, be included as a local library subroutine at the user's installation site. The person responsible for installing or maintaining SINDA'85 at the user's site should be contacted to perform this service.

Location of Subroutine

As a user subroutine, the GETQ subroutine should be included in the user's SINDA'85 model in the SUBROUTINE logic block. The user must enter the text of the subroutine. The GETQ Subroutine has been included as Appendix A. An example of a SINDA'85 model which utilizes GETQ has been included as Appendix B.

If this subroutine will be used often, the user may wish to use the SINDA'85 INCLUDE macroinstruction to simplify reuse.

Location of Call Statement

The GETQ subroutine can be called from any of the SINDA'85 logic blocks (OPERATIONS, VARIABLES 0, VARIABLES 1, VARIABLES 2, OUTPUT CALLS, SUBROUTINE DATA). However, since the temperature values and conductor values are potentially time and temperature dependent, it follows that the heat transfer rate values will represent more realistic values at the end of each solution timestep. Therefore, the most appropriate location for the call to GETQ is in VARIABLES 2 or OUTPUT logic blocks for either transient or steady state analysis.

How to Call GETQ

The call statement to GETQ requires six arguments. The first two arguments are inputs, the remaining four are returned computed heat transfer rate values.

In order, the arguments to GETQ are 1) a character string representing the submodel name for the node of interest; 2) an integer value representing the actual (user assigned) node number; 3) a real variable for storing Q_{i_linear} ; 4) a variable for storing $Q_{i_nonlinear}$; 5) a variable for storing Q_i ; 6) a variable for storing $QLIN+QRAD+QSRC$.

The user may use any properly defined variable names as arguments in the call statement.

DEMONSTRATION OF USE

The GETQ subroutine is very helpful in isolating heat loss from a system into the environment. An illustration of that capability is shown in the following problem description.

Sample Problem Description

A 1 kilogram mass aluminum is heated at a rate of 100 Watts. The exterior surface of the mass has an area of $.01 \text{ m}^2$, an emissivity of $.5$ and a convection coefficient of $10 \text{ W/m}^2\text{-K}$. Assuming a specific heat of 900 J/kg-K , find the heat loss to the environment via radiation and convection separately during the first 30 minutes of heating. Assume the temperature is uniform throughout the mass and that the initial temperature is 100 C . See Figure 1.

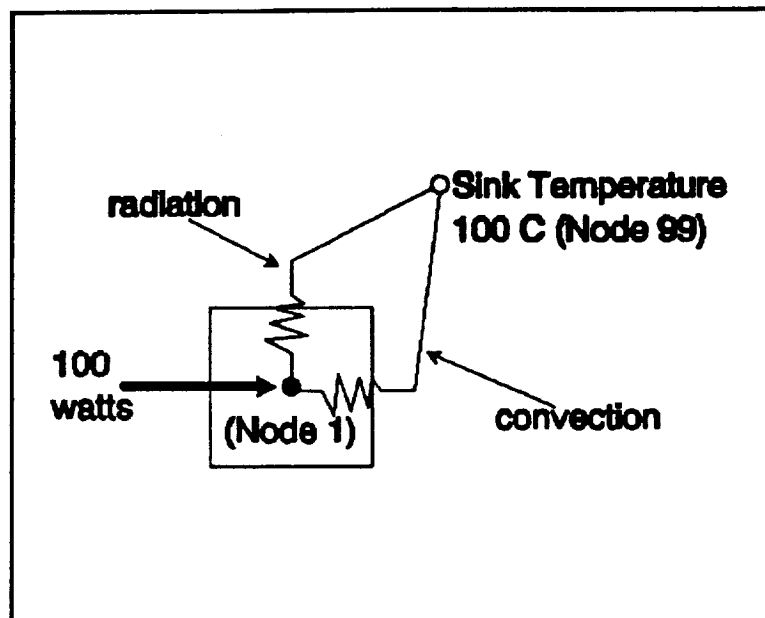


Figure 1. Schematic of Sample Problem

Sample Problem Input File

A SINDA '85 model of the above problem has been included as Appendix B.

The model has a call statement to GETQ in the OUTPUT logic block. Since the desired output is the heat loss to the environment via convection and radiation, the boundary node was chosen as the argument to pass to the GETQ subroutine. For this example, either of the two nodes would be suitable candidates. If, however, the mass were composed of N nodes attached to the boundary node, the boundary node would be the obvious choice of arguments to GETQ. The other option would be to make N calls to GETQ to get the same information.

Once the values of Q_{i_linear} , $Q_{i_nonlinear}$, Q_i , and Q_{i_sum} are returned, the user can use them as desired. In this case, the values are printed to a user file. The user file has been included as Appendix C.

To further illustrate the benefits of this subroutine over the SINDA'85 Library Subroutines, NODMAP was also called from the OUTPUT logic block. NODMAP is one of the many Library Subroutines supplied with SINDA'85. The output from the NODMAP subroutine has been included as Appendix D.

USAGE CAUTIONS

One Way Conductors

One-way conductors in SINDA '85 models are often causes of "ENERGY STABLE BUT UNBALANCED" caution messages generated by the processor. These messages indicate an inability of SINDA '85 to account for the energy flowing out of a node that has a one-way conductor attached. The GETQ Subroutine has the same limitation. Therefore, the Q_{i_linear} term only accounts for the energy through the one-way conductors which have node i as the downstream node.

Using GETQ with DRPMOD

If a user desires to drop a submodel from the current model build, a Library Subroutine, DRPMOD, may be called. The user must be aware that although DRPMOD causes the submodel to be dropped, any conductors which were attached between the remaining built submodels and the dropped submodels will remain active. The temperatures of the dropped submodel will be held constant at the values prior to the DRPMOD call. For such cases, the energy flow calculations performed by GETQ will reflect the temperature and conductance values of the dormant model.

Appendix A - GETQ User Subroutine

```

C-----C
C  SUBROUTINE GETQ(SUBMDL,NODEA,QLIN,QRAD,QSRC,SUMQ)C
C  C
C  C  AUTHOR: DERRICK CHESTONC
C  C  VERSION: 1.0C
C  C  CREATED: OCTOBER 3, 1991C
C  C  UPDATED: AUGUST 15, 1992C
C  C
C  C  FUNCTION: THE SUBROUTINE COMPUTES THE HEAT TRANSFERRED TO A SPECIFIED NODEC
C  C  FROM ALL ATTACHED NODES IN THE CURRENTLY BUILT MODEL, THROUGH C
C  C  ALL CURRENTLY BUILT CONDUCTORS. IT IS TYPICALLY USED TO COMPUTE C
C  C  THE DISTRIBUTION OF HEAT FLOW INTO BOUNDARY NODES.C
C  C
C  C  ARGUMENTS: SUBMDL - THE NAME OF THE SUBMODEL (CHARACTER)C
C  C  NODEA - THE ACTUAL NODE NUMBER FOR WHICH HEAT FLOW IS TO BE C
C  C  COMPUTED (INTEGER)C
C  C  QLIN - RETURNED VALUE OF LINEAR HEAT TRANSFER INTO NODE (REAL)C
C  C  QRAD - RETURNED VALUE OF NONLINEAR HEAT TRANSFER INTO NODE C
C  C  (REAL)C
C  C  QSRC - RETURNED VALUE OF IMPRESSED HEAT LOAD INTO NODE (REAL)C
C  C  SUMQ - SUM OF ABOVE THREE VALUES (REAL)C
C  C
C  C  CALLING: THIS ROUTINE SHOULD BE CALLED FROM VARIABLES 2 OR OUTPUT CALLS C
C-----C
F  SUBROUTINE GETQ(SUBMDL,NODEA,QLIN,QRAD,QSRC,SUMQ)
  CALL COMMON
  CHARACTER SUBMDL*(*)
  REAL*8 QRAD,QLIN,QSRC,SUMQ
  INTEGER NODEA,GOFFST

C*****C
C  FIND RELATIVE NODE NUMBER, NREL FOR NODEA C
C*****C
F  CALL MODTRN(SUBMDL,NODEA,NREL)

C*****C
C  FIND LOCATION OF FIRST LINEAR CONDUCTOR C
C  ATTACHED TO NODEA C
C*****C
F  GOFFST=0
F  DO 10 I=1, NREL-1
F  GOFFST=GOFFST+NLIN(I)+NRAD(I)
F  10 CONTINUE

C*****C
C  INITIALIZE RETURN VARIABLES TO ZERO C
C*****C
F  QLIN=0.0
F  QRAD=0.0
F  SUMQ=0.0

C*****C
C  FOR EACH LINEAR CONDUCTOR ATTACHED TO NODEA C
C  FIND THE CONDUCTANCE VALUE, GAB C
C  FIND THE TEMPERATURE AT THE END, TB C
C  COMPUTE THE HEAT FLOW THROUGH CONDUCTOR, DQ C
C  INCREASE THE QLIN BY DQ C
C  NEXT CONDUCTOR C
C*****C
F  DO 20 I=1,NLIN(NREL)
F  GAB=G(PG(GOFFST+I))
F  TB=T(PG(GOFFST+I))
F  DQ=GAB*(TB-T(NREL))
F  QLIN=QLIN+DQ
F  20 CONTINUE

C*****C
C  FIND LOCATION OF FIRST NONLINEAR CONDUCTOR C
C  ATTACHED TO NODEA C
C*****C
F  GOFFST=GOFFST+NLIN(NREL)

```

```

C*****C
C      FOR EACH NONLINEAR CONDUCTOR ATTACHED TO NODEA C
C      FIND THE CONDUCTANCE VALUE, GAB C
C      FIND THE TEMPERATURE AT THE END, TB C
C      COMPUTE THE HEAT FLOW THROUGH CONDUCTOR, DQ C
C      INCREASE THE QRAD BY DQ C
C      NEXT CONDUCTOR C
C*****C
F      DO 30 I=1,NRAD(NREL)
F      GAB=G(PG(GOFFST+1))
F      TB=T(PT(GOFFST+1))
F      DQ=GAB*SIGMA*((TB-ABSZRO)**4.0 - (T(NREL) -ABSZRO)**4.0)
F      QRAD=QRAD+DQ
F 30 CONTINUE

C*****C
C      COMPUTE IMPRESSED HEAT LOAD ON NODEA, QSRC C
C*****C
F      QSRC=Q(NREL)

C*****C
C      COMPUTE QSUM = QLIN + QSRC + QRAD C
C*****C
F      SUMQ=QLIN+QRAD+QSRC
F      END

```

Appendix B - SINDA '85 Model for Sample Problem

```

HEADER OPTIONS DATA
TITLE SAMPLE PROBLEM DEMONSTRATING GETQ
      OUTPUT=QDOT.OUT
      QMAP=QDOT.MAP
HEADER NODE DATA, BLOCK
  1, 100., 1.*900.
-99, 100., 1.0
HEADER SOURCE DATA, BLOCK
  1, 100.
HEADER CONDUCTOR DATA, BLOCK
-1, 1, 99, .01*.5      $ RADIATION CONDUCTOR AREA*EMIS
  2, 1, 99, .01*10.    $ CONVECTION CONDUCTOR AREA*CONVECTION COEFFICIENT
HEADER USER DATA, GLOBAL
  QAPPLY=1.0      $ VARIABLE FOR STORING IMPRESSED HEAT RATE
  QCONV=1.0      $ VARIABLE FOR STORING CONVECTION HEAT TRANSFER RATE
  QNLIN=1.0      $ VARIABLE FOR STORING RADIATION HEAT TRANSFER RATE
  QTOTL=1.0      $ VARIABLE FOR STORING TOTAL HEAT TRANSFER RATE
HEADER CONTROL DATA, GLOBAL
  SIGMA=5.6778E-8
  ABSZRO=-273.15
  TIMEN=60.*30.    $ STOP SOLUTION AT 30 MINUTES
  OUTPUT=30.       $ OUTPUT INTERVAL 30 SECONDS
HEADER OPERATIONS DATA
BUILD ASSBL,BLOCK
  CALL FLDLCK
HEADER OUTPUT CALLS, BLOCK
  CALL GETQ('BLOCK',99,QCONV,QNLIN,QAPPLY,QTOTL)
  WRITE(71,100) TIMEN/60., QCONV, QNLIN
100 FORMAT('TIME (MINUTES) = ', F10.4,5X,
+         'CONVECTION HEAT LOSS (WATTS)=',E12.5,5X,
+         'RADIATION HEAT LOSS (WATTS)=', E12.5)
  CALL MODMAP('BLOCK',99,1)
  CALL TPRINT('ALL')
HEADER SUBROUTINE DATA
-----C-----
C  SUBROUTINE GETQ(SUBMDL,NODEA,QLIN,GRAD,QSRC,SUMQ)
C
C  AUTHOR: DERRICK CHESTON
C  VERSION: 1.0
C  CREATED: OCTOBER 3, 1991
C  UPDATED: AUGUST 15, 1992
C
C  FUNCTION: THE SUBROUTINE COMPUTES THE HEAT TRANSFERRED TO A SPECIFIED NODE
C            FROM ALL ATTACHED NODES IN THE CURRENTLY BUILT MODEL, THROUGH
C            ALL CURRENTLY BUILT CONDUCTORS. IT IS TYPICALLY USED TO COMPUTE
C            THE DISTRIBUTION OF HEAT FLOW INTO BOUNDARY NODES.
C
C  ARGUMENTS: SUBMDL - THE NAME OF THE SUBMODEL (CHARACTER)
C            NODEA   - THE ACTUAL NODE NUMBER FOR WHICH HEAT FLOW IS TO BE
C                     COMPUTED (INTEGER)
C            QLIN    - RETURNED VALUE OF LINEAR HEAT TRANSFER INTO NODE (REAL)
C            GRAD    - RETURNED VALUE OF NONLINEAR HEAT TRANSFER INTO NODE
C                     (REAL)
C            QSRC    - RETURNED VALUE OF IMPRESSED HEAT LOAD INTO NODE (REAL)
C            SUMQ    - SUM OF ABOVE THREE VALUES (REAL)
C
C  CALLING: THIS ROUTINE SHOULD BE CALLED FROM VARIABLES 2 OR OUTPUT CALLS
C-----C-----
F  SUBROUTINE GETQ(SUBMDL,NODEA,QLIN,GRAD,QSRC,SUMQ)
F    CALL COMMON
F    CHARACTER SUBMDL*(*)
F    REAL*8 GRAD,QLIN,QSRC,SUMQ
F    INTEGER NODEA,GOFFST
C*****C
C    FIND RELATIVE NODE NUMBER, NREL FOR NODEA  C
C*****C
F    CALL MODTRN(SUBMDL,NODEA,NREL)
C*****C
C    FIND LOCATION OF FIRST LINEAR CONDUCTOR  C
C    ATTACHED TO NODEA                        C

```

```

C*****C
F      GOFFST=0
F      DO 10 I=1, NREL-1
F          GOFFST=GOFFST+NLIN(I)+NRAD(I)
F  10 CONTINUE

C*****C
C      INITIALIZE RETURN VARIABLES TO ZERO      C
C*****C
F      QLIN=0.0
F      GRAD=0.0
F      QSUM=0.0

C*****C
C      FOR EACH LINEAR CONDUCTOR ATTACHED TO NODEA      C
C          FIND THE CONDUCTANCE VALUE, GAB      C
C          FIND THE TEMPERATURE AT THE END, TB      C
C          COMPUTE THE HEAT FLOW THROUGH CONDUCTOR, DQ      C
C          INCREASE THE QLIN BY DQ      C
C      NEXT CONDUCTOR      C
C*****C
F      DO 20 I=1,NLIN(NREL)
F          GAB=G(PG(GOFFST+I))
F          TB=T(PT(GOFFST+I))
F          DQ=GAB*(TB-T(NREL))
F          QLIN=QLIN+DQ
F  20 CONTINUE

C*****C
C      FIND LOCATION OF FIRST NONLINEAR CONDUCTOR      C
C      ATTACHED TO NODEA      C
C*****C
F      GOFFST=GOFFST+NLIN(NREL)

C*****C
C      FOR EACH NONLINEAR CONDUCTOR ATTACHED TO NODEA      C
C          FIND THE CONDUCTANCE VALUE, GAB      C
C          FIND THE TEMPERATURE AT THE END, TB      C
C          COMPUTE THE HEAT FLOW THROUGH CONDUCTOR, DQ      C
C          INCREASE THE GRAD BY DQ      C
C      NEXT CONDUCTOR      C
C*****C
F      DO 30 I=1,NRAD(NREL)
F          GAB=G(PG(GOFFST+I))
F          TB=T(PT(GOFFST+I))
F          DQ=GAB*SIGNA*((TB-ABSZRO)**4.0 - (T(NREL) -ABSZRO)**4.0)
F          GRAD=GRAD+DQ
F  30 CONTINUE

C*****C
C      COMPUTE IMPRESSED HEAT LOAD ON NODEA, QSRC      C
C*****C
F      QSRC=Q(NREL)

C*****C
C      COMPUTE QSUM = QLIN + QSRC + GRAD      C
C*****C
F      QSUM=QLIN+GRAD+QSRC
F      END
F      END OF DATA

```

C-4

Appendix C - GETQ Output from Sample Problem

TIME (MINUTES) =	0.0000	CONVECTION HEAT LOSS (WATTS)=	0.00000E+00	RADIATION HEAT LOSS (WATTS)=	0.00000E+00
TIME (MINUTES) =	0.5000	CONVECTION HEAT LOSS (WATTS)=	0.33244E+00	RADIATION HEAT LOSS (WATTS)=	0.19878E+00
TIME (MINUTES) =	1.0000	CONVECTION HEAT LOSS (WATTS)=	0.66311E+00	RADIATION HEAT LOSS (WATTS)=	0.40180E+00
TIME (MINUTES) =	1.5000	CONVECTION HEAT LOSS (WATTS)=	0.99200E+00	RADIATION HEAT LOSS (WATTS)=	0.60905E+00
TIME (MINUTES) =	2.0000	CONVECTION HEAT LOSS (WATTS)=	0.13191E+01	RADIATION HEAT LOSS (WATTS)=	0.82053E+00
TIME (MINUTES) =	2.5000	CONVECTION HEAT LOSS (WATTS)=	0.16444E+01	RADIATION HEAT LOSS (WATTS)=	0.10363E+01
TIME (MINUTES) =	3.0000	CONVECTION HEAT LOSS (WATTS)=	0.19679E+01	RADIATION HEAT LOSS (WATTS)=	0.12562E+01
TIME (MINUTES) =	3.5000	CONVECTION HEAT LOSS (WATTS)=	0.22896E+01	RADIATION HEAT LOSS (WATTS)=	0.14804E+01
TIME (MINUTES) =	4.0000	CONVECTION HEAT LOSS (WATTS)=	0.26094E+01	RADIATION HEAT LOSS (WATTS)=	0.17087E+01
TIME (MINUTES) =	4.5000	CONVECTION HEAT LOSS (WATTS)=	0.29274E+01	RADIATION HEAT LOSS (WATTS)=	0.19413E+01
TIME (MINUTES) =	5.0000	CONVECTION HEAT LOSS (WATTS)=	0.32436E+01	RADIATION HEAT LOSS (WATTS)=	0.21781E+01
TIME (MINUTES) =	5.5000	CONVECTION HEAT LOSS (WATTS)=	0.35580E+01	RADIATION HEAT LOSS (WATTS)=	0.24190E+01
TIME (MINUTES) =	6.0000	CONVECTION HEAT LOSS (WATTS)=	0.38704E+01	RADIATION HEAT LOSS (WATTS)=	0.26641E+01
TIME (MINUTES) =	6.5000	CONVECTION HEAT LOSS (WATTS)=	0.41811E+01	RADIATION HEAT LOSS (WATTS)=	0.29133E+01
TIME (MINUTES) =	7.0000	CONVECTION HEAT LOSS (WATTS)=	0.44898E+01	RADIATION HEAT LOSS (WATTS)=	0.31666E+01
TIME (MINUTES) =	7.5000	CONVECTION HEAT LOSS (WATTS)=	0.47967E+01	RADIATION HEAT LOSS (WATTS)=	0.34241E+01
TIME (MINUTES) =	8.0000	CONVECTION HEAT LOSS (WATTS)=	0.51017E+01	RADIATION HEAT LOSS (WATTS)=	0.36855E+01
TIME (MINUTES) =	8.5000	CONVECTION HEAT LOSS (WATTS)=	0.54048E+01	RADIATION HEAT LOSS (WATTS)=	0.39510E+01
TIME (MINUTES) =	9.0000	CONVECTION HEAT LOSS (WATTS)=	0.57060E+01	RADIATION HEAT LOSS (WATTS)=	0.42205E+01
TIME (MINUTES) =	9.5000	CONVECTION HEAT LOSS (WATTS)=	0.60052E+01	RADIATION HEAT LOSS (WATTS)=	0.44939E+01
TIME (MINUTES) =	10.0000	CONVECTION HEAT LOSS (WATTS)=	0.63026E+01	RADIATION HEAT LOSS (WATTS)=	0.47713E+01
TIME (MINUTES) =	10.5000	CONVECTION HEAT LOSS (WATTS)=	0.65981E+01	RADIATION HEAT LOSS (WATTS)=	0.50526E+01
TIME (MINUTES) =	11.0000	CONVECTION HEAT LOSS (WATTS)=	0.68916E+01	RADIATION HEAT LOSS (WATTS)=	0.53377E+01
TIME (MINUTES) =	11.5000	CONVECTION HEAT LOSS (WATTS)=	0.71832E+01	RADIATION HEAT LOSS (WATTS)=	0.56266E+01
TIME (MINUTES) =	12.0000	CONVECTION HEAT LOSS (WATTS)=	0.74729E+01	RADIATION HEAT LOSS (WATTS)=	0.59193E+01
TIME (MINUTES) =	12.5000	CONVECTION HEAT LOSS (WATTS)=	0.77606E+01	RADIATION HEAT LOSS (WATTS)=	0.62156E+01
TIME (MINUTES) =	13.0000	CONVECTION HEAT LOSS (WATTS)=	0.80464E+01	RADIATION HEAT LOSS (WATTS)=	0.65157E+01
TIME (MINUTES) =	13.5000	CONVECTION HEAT LOSS (WATTS)=	0.83302E+01	RADIATION HEAT LOSS (WATTS)=	0.68193E+01
TIME (MINUTES) =	14.0000	CONVECTION HEAT LOSS (WATTS)=	0.86120E+01	RADIATION HEAT LOSS (WATTS)=	0.71265E+01
TIME (MINUTES) =	14.5000	CONVECTION HEAT LOSS (WATTS)=	0.88919E+01	RADIATION HEAT LOSS (WATTS)=	0.74372E+01
TIME (MINUTES) =	15.0000	CONVECTION HEAT LOSS (WATTS)=	0.91698E+01	RADIATION HEAT LOSS (WATTS)=	0.77514E+01
TIME (MINUTES) =	15.5000	CONVECTION HEAT LOSS (WATTS)=	0.94458E+01	RADIATION HEAT LOSS (WATTS)=	0.80690E+01
TIME (MINUTES) =	16.0000	CONVECTION HEAT LOSS (WATTS)=	0.97197E+01	RADIATION HEAT LOSS (WATTS)=	0.83899E+01
TIME (MINUTES) =	16.5000	CONVECTION HEAT LOSS (WATTS)=	0.99917E+01	RADIATION HEAT LOSS (WATTS)=	0.87140E+01
TIME (MINUTES) =	17.0000	CONVECTION HEAT LOSS (WATTS)=	0.10262E+02	RADIATION HEAT LOSS (WATTS)=	0.90414E+01
TIME (MINUTES) =	17.5000	CONVECTION HEAT LOSS (WATTS)=	0.10530E+02	RADIATION HEAT LOSS (WATTS)=	0.93719E+01
TIME (MINUTES) =	18.0000	CONVECTION HEAT LOSS (WATTS)=	0.10796E+02	RADIATION HEAT LOSS (WATTS)=	0.97055E+01
TIME (MINUTES) =	18.5000	CONVECTION HEAT LOSS (WATTS)=	0.11060E+02	RADIATION HEAT LOSS (WATTS)=	0.10042E+02
TIME (MINUTES) =	19.0000	CONVECTION HEAT LOSS (WATTS)=	0.11322E+02	RADIATION HEAT LOSS (WATTS)=	0.10382E+02
TIME (MINUTES) =	19.5000	CONVECTION HEAT LOSS (WATTS)=	0.11582E+02	RADIATION HEAT LOSS (WATTS)=	0.10724E+02
TIME (MINUTES) =	20.0000	CONVECTION HEAT LOSS (WATTS)=	0.11840E+02	RADIATION HEAT LOSS (WATTS)=	0.11069E+02
TIME (MINUTES) =	20.5000	CONVECTION HEAT LOSS (WATTS)=	0.12096E+02	RADIATION HEAT LOSS (WATTS)=	0.11417E+02
TIME (MINUTES) =	21.0000	CONVECTION HEAT LOSS (WATTS)=	0.12350E+02	RADIATION HEAT LOSS (WATTS)=	0.11768E+02
TIME (MINUTES) =	21.5000	CONVECTION HEAT LOSS (WATTS)=	0.12601E+02	RADIATION HEAT LOSS (WATTS)=	0.12121E+02
TIME (MINUTES) =	22.0000	CONVECTION HEAT LOSS (WATTS)=	0.12851E+02	RADIATION HEAT LOSS (WATTS)=	0.12476E+02
TIME (MINUTES) =	22.5000	CONVECTION HEAT LOSS (WATTS)=	0.13099E+02	RADIATION HEAT LOSS (WATTS)=	0.12834E+02
TIME (MINUTES) =	23.0000	CONVECTION HEAT LOSS (WATTS)=	0.13345E+02	RADIATION HEAT LOSS (WATTS)=	0.13195E+02
TIME (MINUTES) =	23.5000	CONVECTION HEAT LOSS (WATTS)=	0.13589E+02	RADIATION HEAT LOSS (WATTS)=	0.13558E+02
TIME (MINUTES) =	24.0000	CONVECTION HEAT LOSS (WATTS)=	0.13831E+02	RADIATION HEAT LOSS (WATTS)=	0.13922E+02
TIME (MINUTES) =	24.5000	CONVECTION HEAT LOSS (WATTS)=	0.14071E+02	RADIATION HEAT LOSS (WATTS)=	0.14289E+02
TIME (MINUTES) =	25.0000	CONVECTION HEAT LOSS (WATTS)=	0.14308E+02	RADIATION HEAT LOSS (WATTS)=	0.14658E+02
TIME (MINUTES) =	25.5000	CONVECTION HEAT LOSS (WATTS)=	0.14544E+02	RADIATION HEAT LOSS (WATTS)=	0.15029E+02
TIME (MINUTES) =	26.0000	CONVECTION HEAT LOSS (WATTS)=	0.14778E+02	RADIATION HEAT LOSS (WATTS)=	0.15402E+02
TIME (MINUTES) =	26.5000	CONVECTION HEAT LOSS (WATTS)=	0.15010E+02	RADIATION HEAT LOSS (WATTS)=	0.15776E+02
TIME (MINUTES) =	27.0000	CONVECTION HEAT LOSS (WATTS)=	0.15239E+02	RADIATION HEAT LOSS (WATTS)=	0.16152E+02
TIME (MINUTES) =	27.5000	CONVECTION HEAT LOSS (WATTS)=	0.15467E+02	RADIATION HEAT LOSS (WATTS)=	0.16530E+02
TIME (MINUTES) =	28.0000	CONVECTION HEAT LOSS (WATTS)=	0.15693E+02	RADIATION HEAT LOSS (WATTS)=	0.16909E+02
TIME (MINUTES) =	28.5000	CONVECTION HEAT LOSS (WATTS)=	0.15916E+02	RADIATION HEAT LOSS (WATTS)=	0.17290E+02
TIME (MINUTES) =	29.0000	CONVECTION HEAT LOSS (WATTS)=	0.16138E+02	RADIATION HEAT LOSS (WATTS)=	0.17672E+02
TIME (MINUTES) =	29.5000	CONVECTION HEAT LOSS (WATTS)=	0.16358E+02	RADIATION HEAT LOSS (WATTS)=	0.18055E+02
TIME (MINUTES) =	30.0000	CONVECTION HEAT LOSS (WATTS)=	0.16575E+02	RADIATION HEAT LOSS (WATTS)=	0.18440E+02

Appendix D - NODMAP Output from Sample Problem

A MAP OF INPUT BOUND NODE BLOCK 99 (INTERNAL 2)

THE PARAMETERS OF NODE BLOCK 99 ARE:

TEMPERATURE = 100.000 (DEG.)

CAPACITANCE = 1.00000 (ENERGY/DEG)

NET SOURCE/SINK = 0.000000E+00 (ENERGY/TIME, INCLUDES TIES)

CAP./SUM OF COND. = 4.73377 (TIME, INCLUDES TIES)

THE ADJOINING NODES TO NODE BLOCK 99 ARE:

MODE INPUT	CONDUCTOR (INTERNAL) INPUT (INTERNAL)	TYPE	CONDUCTOR VALUE	% OF TYPE	% OF TOTAL	HEAT TRANSFER RATE (ENERGY/TIME)	TEMPERATURE OF ADJOINING NODE
BLOCK	1(1) 2 (1)	LINEAR	0.100000	100.0	47.3	16.5753	265.753
BLOCK	1(1) 1 (2)	RADIAT	5.000000E-03	100.0	52.7	18.4397	265.753

THE TOTALS ON NODE BLOCK 99 ARE:

LINEAR HEAT TRANSFER (CONDUCTION/CONVECTION)... 16.5753

RADIATION HEAT TRANSFER..... 18.4397

HEAT SOURCE/SINKS APPLIED..... 0.000000E+00

35.0150 (ENERGY/TIME)

EFFECTIVE ERN TEMPERATURE..... 265.753

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1992	3. REPORT TYPE AND DATES COVERED Conference Publication		
4. TITLE AND SUBTITLE Fourth Annual Thermal and Fluids Analysis Workshop		5. FUNDING NUMBERS		
6. AUTHOR(S)				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		8. PERFORMING ORGANIZATION REPORT NUMBER E-7346		
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CP-10106		
11. SUPPLEMENTARY NOTES Responsible person, Doug Darling, (216) 433-8115.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories 34, 61, and 64			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Fourth Annual Thermal and Fluids Analysis Workshop was held from August 17-21, 1992, at NASA Lewis Research Center. The workshop consisted of classes, vendor demonstrations, and paper sessions. The classes and vendor demonstrations provided participants with the information on widely used tools for thermal and fluids analysis. The paper sessions provided a forum for the exchange of information and ideas among thermal and fluids analysts. Paper topics included advances and uses of established thermal and fluids computer codes (such as SINDA and TRASYS) as well as unique modeling techniques and applications.				
14. SUBJECT TERMS Thermal simulation; Fluid mechanics; Computer programs			15. NUMBER OF PAGES 289	
			16. PRICE CODE A13	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	